INVESTIGATION OF PROBABILISTIC PRINCIPAL COMPONENT ANALYSIS COMPARED TO PROPER ORTHOGONAL DECOMPOSITION METHODS FOR BASIS EXTRACTION AND MISSING DATA ESTIMATION

A Dissertation
Presented to
The Academic Faculty

by

Kyunghoon Lee

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Aerospace Engineering

Georgia Institute of Technology
August 2010

INVESTIGATION OF PROBABILISTIC PRINCIPAL COMPONENT ANALYSIS COMPARED TO PROPER ORTHOGONAL DECOMPOSITION METHODS FOR BASIS EXTRACTION AND MISSING DATA ESTIMATION

Approved by:

Professor Dimitri N. Mavris, Advisor
School of Aerospace Engineering
Georgia Institute of Technology

Professor Brian German
School of Aerospace Engineering
Georgia Institute of Technology

Professor Olivier Bauchau
School of Aerospace Engineering
Georgia Institute of Technology

Professor Karen E. Willcox
Department of Aeronautics and Astronautics
Massachusetts Institute of Technology

Professor Lakshmi N. Sankar
School of Aerospace Engineering
Georgia Institute of Technology

Date Approved: May 2010

To my parents and my little sister,

who have always believed in me,

and

to my friends,

who have stood by me through thick and thin

ACKNOWLEDGEMENTS

Ever since an aircraft aloft in the air mesmerized me with its transcendental magnificence, I have strived to comprehend all of its mysterious splendor and majesty bestowed upon it. I believe my tenacious passion and admiration for aircraft have kept me motivated, eventually guiding me to doctoral study in the realm of aerospace system design.

In pursuing a Ph.D. degree in the Aerospace Systems Design Laboratory (ASDL) at Georgia Tech, I am gratefully indebted to many people who have helped me throughout my study. First and foremost, I would not have been able to make it through without the support and nurturing of my advisor, Professor Dimitri N. Mavris. He opened my eyes to probabilistic aircraft design approaches when I was biased to optimization-driven design, ultimately changing my mindset for aircraft system design. I would also like to extend my deepest gratitude to my other committee members. To begin with, I have learned a lot from Professor Olivier A. Bauchau as I was inundated with his challenging homework assignments and exams—but I really enjoyed every aspect of his classes. He always showed ingenious insights in addressing engineering problems from a physics-based perspective without being overwhelmed by the recondite mathematical theories behind them. Professor Lakshmi N. Sankar was generous with his potential flow solvers and willing to provide any help if necessary for my thesis research. Professor Brian German, a former alumnus of ASDL, contributed beneficial feedback from an aircraft system-oriented viewpoint. Last but not least, Professor Karen E. Willcox from MIT gladly agreed to be the last member of my thesis committee, and her outstanding work on gappy proper orthogonal decomposition (POD) closely pertain to my thesis research.

I cannot leave the Georgia Institute of Technology without expressing my sincere thanks to those who cooperated with me in the process of my thesis research. First off, Dr. Taewoo Nam suggested a valuable application of my thesis idea for the reduced-order modeling of the numerical propulsion system simulation (NPSS). He was also considerate enough to

regularly check my thesis work so that I could continue to make headway in accordance with my thesis schedule. In connection with the reduced-order modeling of NPSS, Mr. Christopher Perullo collaborated with us by devising an NPSS engine model and generating engine simulation data. Similarly, the other application in my thesis, dealing with particle image velocimetry (PIV) data, benefited from Professor Timothy C. Lieuwen and his student Mr. Dong-Hyuk Shin; the former was willing to allow me access to PIV data collected in his laboratory, and the latter never wavered in his support in processing the PIV data. Thanks should also go to Dr. Byung-Young Min for his computational fluid dynamics (CFD) solver, which I have heavily relied upon to generate test sample data sets for my thesis research. Dr. Hernando Jimenez, a master of formulating hypotheses and research questions, deserves my thanks for helping me reorganize my hypotheses and research questions in a clear, logical way.

While writing up my thesis, I received invaluable contributions from several individuals. From a technical aspect, I was able to enrich the soundness and coherence of my thesis due to constructive remarks from the following colleagues: Dr. Jongki Moon, Dr. Dongwook Lim, Dr. Young Ki Lee, and Mr. Kyungjin Moon. I also would like to acknowledge comments and suggestions from anonymous referees when I submitted some of my thesis work to refereed journals. In addition to those technical reviews, my thesis would not have been grammatically acceptable and readable if I had not abundantly consulted with Jane Chisholm. She literally cultivated my writing style in view of the reader's perspective; most parts of my thesis have already gone through her eyes—this acknowledgement chapter is no exception. Along with Jane, Nonnie Kim willingly squeezed her time to proofread a few chapters of my thesis. In the process of this thesis writing, LaTeX and gnuplot were instrumental in typesetting and plotting; I am grateful to countless contributors to such valuable and praiseworthy software. As Rage Against the Machine prided themselves on producing all songs with no artificial sounds, I am pleased that I utilized only LaTeX and gnuplot except for a few figures that required my use of Tecplot.

As I look back upon those days when I started my graduate study, I cannot begin to show my appreciation to the many individuals who helped me settle down in Atlanta.

Without the financial support of my uncle, Young-Sun Hwang, I wound not have been able to entertain the idea of studying abroad. Once I arrived here in Atlanta, I received tremendous aid from numerous people, especially Dr. Donghoon Lee and Dr. Hyungjoo Yoon, until I finally became settled. Back in South Korea for vacation, it was always a great pleasure for me to catch up with my friends in the class of 2001 at Pusan National University (PNU). Of course, I would not have been able to survive my Ph.D. study under so much stress without playing tennis for R & R. I would like to thank my tennis buddies, Dong-Hyuk Shin and Yong-Jae Kim, as well as people in the tennis club of the Georgia Tech Korean Student Association. Thanks also go to my other tennis partners in the Volcanoes Tennis Club, who played in Atlanta Lawn Tennis Association (ATLA) matches with me. Although my probability of winning has not been very high, I'm getting the hang of it, and my game should improve slowly but steadily. Finally, my parents and my little sister deserve my sincere gratitude for their relentless support and their profound belief in me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ASDL	Aerospace Systems Design Laboratory
BADA	base of aircraft data
CFD	computational fluid dynamics
DNS	direct numerical simulation
DoE	design of experiments
DPPCA	dual probabilistic principal component analysis
EM	expectation-maximization
EM-PCA	EM algorithm for PPCA
EINOx	emission index NO_X
E-step	expectation step
EVD	eigenvalue decomposition
FDS	flux difference splitting
FPE	full potential equation
FPR	fan pressure ratio
GENCAS	generic numerical compressible airflow solver
HPCPR	high-pressure compressor pressure ratio
KLT	Karhunen-Loève transform
LHD	Latin hypercube design
LHS	left-hand side

LPCPR	low-pressure compressor pressure ratio
LU-SGS	lower-upper symmetric Gauss-Seidel
MaxT41	maximum turbine inlet temperature
MLE	maximum likelihood estimate
MRI	magnetic resonance imaging
MSE	mean square error
M-step	maximization step
MUSCL	monotone upstream-centered schemes for conservation laws
MWR	method of weighted residuals
NPSS	numerical propulsion system simulation
NRMSE	normalized root mean square error
NRSE	normalized root square error
ODE	ordinary differential equation
PCA	principal component analysis
PDE	partial differential equation
PIV	particle image velocimetry
POD	proper orthogonal decomposition
PPCA	probabilistic principal component analysis
RBF	radial basis function
RHS	right-hand side
RMSE	root mean square error

RMSR	root mean square residual
ROM	reduced-order modeling
RSM	response surface methodology
R^2	coefficient of determination
SLST	sea-level static thrust
SVD	singular value decomposition
SVM	support vector machine

NOMENCLATURE

$\mathbb{R}^n$	n-dimensional real number space
$\mathcal{L}$	log-likelihood function
$\mathcal{L}_C$	complete-data log-likelihood function
$\mathcal{N}$	Gaussian probability distribution
$\mathbf{0}$	zero matrix
$\mathbf{C}$	model covariance matrix
$\mathbf{I}$	identity matrix
$\mathbf{S}$	sample covariance matrix
$\mathbf{T}$	collection of error-accounted observed variables
$\mathbf{V}$	eigenvector matrix, i.e., a POD basis
$\mathbf{V}_e$	guessed $\mathbf{V}_q$ obtained from $\widetilde{\mathbf{Y}}^{(0)}$
$\mathbf{V}_q$	matrix of the first q eigenvectors
$\mathbf{W}$	factor-loading matrix
$\mathbf{X}$	collection of latent variables
$\mathbf{Y}$	collection of observed variables
$\mathbf{1}_N$	vector with N ones
$\mathbf{b}$	least-squares coefficient of gappy POD
$\mathbf{c}$	generalized least-squares coefficient
$\mathbf{n}$	mask vector

$\mathbf{t}$	error-accounted observed variable
$\mathbf{x}$	latent variable
$\mathbf{y}$	observed variable
d	dimension of an observed variable
k	number of iterations
N	snapshot ensemble size
p	probability density function
q	dimension of a latent variable, i.e., model selection
R	averaged estimation error
r	estimation error
s	number of data-missing snapshots

Subscripts

ML	maximum likelihood estimate
j	j^{th} vector

Conventions

$(\,\cdot\,,\,\cdot\,)$	inner product
$\circ$	Hadamard product, i.e., point-wise multiplication
$\dot{(\cdot)}$	mean-centered data
$\langle\,\cdot\,\rangle$	expectation
$\|\cdot\|_{L^2}$	L^2 norm
$\|\cdot\|_n$	gappy norm

$\mathcal{O}(\,\cdot\,)$	big O notation
$(\,\mathring{\cdot}\,)$	data with missing values
$\mathrm{diag}(\,\cdot\,)$	diagonals of matrix
$\mathrm{tr}(\,\cdot\,)$	trace
w.r.t.	with respect to
$\widetilde{(\,\cdot\,)}$	estimation

Symbols

α	general norm
λ	eigenvalue
$\boldsymbol{\epsilon}$	error variable
$\boldsymbol{\Lambda}$	eigenvalue matrix
$\boldsymbol{\Omega}$	diagonal matrix
$\boldsymbol{\Phi}$	basis
$\mathbf{Q}$	orthogonal matrix
σ^2	variance
$\boldsymbol{\mu}$	mean vector

Superscripts

(k)	k^{th} iteration

SUMMARY

The identification of flow characteristics and the reduction of high-dimensional simulation data have capitalized on an orthogonal basis achieved by proper orthogonal decomposition (POD), also known as principal component analysis (PCA) or the Karhunen-Loève transform (KLT). In the realm of aerospace engineering, an orthogonal basis is versatile for diverse applications, especially associated with reduced-order modeling (ROM) as follows: a low-dimensional turbulence model, an unsteady aerodynamic model for aeroelasticity and flow control, and a steady aerodynamic model for airfoil shape design. Provided that a given data set lacks parts of its data, POD is required to adopt a least-squares formulation, leading to gappy POD, using a gappy norm that is a variant of an L^2 norm dealing with only known data. Although gappy POD is originally devised to restore marred images, its application has spread to aerospace engineering for the following reason: various engineering problems can be reformulated in forms of missing data estimation to exploit gappy POD. Similar to POD, gappy POD has a broad range of applications such as optimal flow sensor placement, experimental and numerical flow data assimilation, and impaired particle image velocimetry (PIV) data restoration.

Apart from POD and gappy POD, both of which are deterministic formulations, probabilistic principal component analysis (PPCA), a probabilistic generalization of PCA, has been used in the pattern recognition field for speech recognition and in the oceanography area for empirical orthogonal functions in the presence of missing data. In formulation, PPCA presumes a linear latent variable model relating an observed variable with a latent variable that is inferred only from an observed variable through a linear mapping called factor-loading. To evaluate the maximum likelihood estimates (MLEs) of PPCA parameters such as a factor-loading, PPCA can invoke an expectation-maximization (EM) algorithm, yielding an EM algorithm for PPCA (EM-PCA). By virtue of the EM algorithm, the EM-PCA is capable of not only extracting a basis but also restoring missing data through

iterations whether the given data are intact or not. Therefore, the EM-PCA can potentially substitute for both POD and gappy POD inasmuch as its accuracy and efficiency are comparable to those of POD and gappy POD. In order to examine the benefits of the EM-PCA for aerospace engineering applications, this thesis attempts to qualitatively and quantitatively scrutinize the EM-PCA alongside both POD and gappy POD using high-dimensional simulation data.

In pursuing qualitative investigations, the theoretical relationship between POD and PPCA is transparent such that the factor-loading MLE of PPCA, evaluated by the EM-PCA, pertains to an orthogonal basis obtained by POD. By contrast, the analytical connection between gappy POD and the EM-PCA is nebulous because they distinctively approximate missing data due to their antithetical formulation perspectives: gappy POD solves a least-squares problem whereas the EM-PCA relies on the expectation of the observation probability model. To juxtapose both gappy POD and the EM-PCA, this research proposes a unifying least-squares perspective that embraces the two disparate algorithms within a generalized least-squares framework. As a result, the unifying perspective reveals that both methods address similar least-squares problems; however, their formulations contain dissimilar bases and norms. Furthermore, this research delves into the ramifications of the different bases and norms that will eventually characterize the traits of both methods. To this end, two hybrid algorithms of gappy POD and the EM-PCA are devised and compared to the original algorithms for a qualitative illustration of the different basis and norm effects. After all, a norm reflecting a curve-fitting method is found to more significantly affect estimation error reduction than a basis for two example test data sets: one is absent of data only at a single snapshot and the other misses data across all the snapshots.

From a numerical performance aspect, the EM-PCA is computationally less efficient than POD for intact data since it suffers from slow convergence inherited from the EM algorithm. For incomplete data, this thesis quantitatively found that the number of data-missing snapshots predetermines whether the EM-PCA or gappy POD outperforms the other because of the computational cost of a coefficient evaluation, resulting from a norm selection. For instance, gappy POD demands laborious computational effort in proportion

to the number of data-missing snapshots as a consequence of the gappy norm. In contrast, the computational cost of the EM-PCA is invariant to the number of data-missing snapshots thanks to the L^2 norm. In general, the higher the number of data-missing snapshots, the wider the gap between the computational cost of gappy POD and the EM-PCA. Based on the numerical experiments reported in this thesis, the following criterion is recommended regarding the selection between gappy POD and the EM-PCA for computational efficiency: gappy POD for an incomplete data set containing a few data-missing snapshots and the EM-PCA for an incomplete data set involving multiple data-missing snapshots.

Last, the EM-PCA is applied to two aerospace applications in comparison to gappy POD as a proof of concept: one with an emphasis on basis extraction and the other with a focus on missing data reconstruction for a given incomplete data set with scattered missing data. The first application exploits the EM-PCA to efficiently construct reduced-order models of engine deck responses obtained by the numerical propulsion system simulation (NPSS), some of whose results are absent due to failed analyses caused by numerical instability. Model-prediction tests validate that engine performance metrics estimated by the reduced-order NPSS model exhibit considerably good agreement with those directly obtained by NPSS. Similarly, the second application illustrates that the EM-PCA is significantly more cost effective than gappy POD at repairing spurious PIV measurements obtained from acoustically-excited, bluff-body jet flow experiments. The EM-PCA reduces computational cost on factors $8 \sim 19$ compared to gappy POD while generating the same restoration results as those evaluated by gappy POD. All in all, through comprehensive theoretical and numerical investigation, this research establishes that the EM-PCA is an efficient alternative to gappy POD for an incomplete data set containing missing data over an entire data set.

CHAPTER I

INTRODUCTION

1.1 Motivation

Due to empowering computing environments and enhanced numerical schemes, those involved in design-oriented analyses for complex systems such as aircraft have been attempting to exploit high-fidelity, physics-based simulation tools. Although low- and moderate-fidelity models are computationally inexpensive, their results are reliable mostly only for either conventional or uncomplicated applications. As a result, in such design studies, the benefits of simplified models, whose assumptions and heuristics are fragile, quickly disappear. For instance, aircraft designs seeking to mitigate fuel consumption and noise are apt to achieve their goals through dramatic shape changes that are beyond evolutionary improvements of conventional configurations. Furthermore, the current physics-based design trends necessitate high-fidelity simulations that infuse more accurate information into the early design phase for lower design turn-around time and development costs. Therefore, high-fidelity analyses are indispensable to broadening the horizon of aircraft system design capabilities in terms of accuracy and application generality.

Nonetheless, in multidisciplinary analysis and design studies, numerical simulations based on low- or moderate-fidelity models are still prevalent mainly because of their practicality. A high-fidelity simulation is often cumbersome to utilize because it requires not only arduous integration work but also enormous computational time and resources. The use of a high-fidelity analysis requires even more effort in a nondeterministic or optimization-driven design study, each of which demands a colossal number of simulations during the design process. As an illustration, Figure 1 delineates the characteristics of both high- and low-fidelity analyses, contrasting their strengths and weaknesses in four criteria: accuracy, application generality, integration difficulty, and computational speed. In Figure 1, the accuracy of a low-fidelity analysis is denoted with dashed and solid lines; the former represents

a case in which the assumptions of a low-fidelity analysis are legitimate; otherwise, the latter indicates the accuracy of a low-fidelity analysis. Note that the benefits of a high-fidelity analysis are inversely the drawbacks of a low-fidelity analysis, and vice versa. As conveyed in Figure 1, to facilitate the use of a high-fidelity analysis for design studies, one would like to rely on a compromise that accommodates all the four different virtues of both high- and low-fidelity analyses, indicated with a green line.

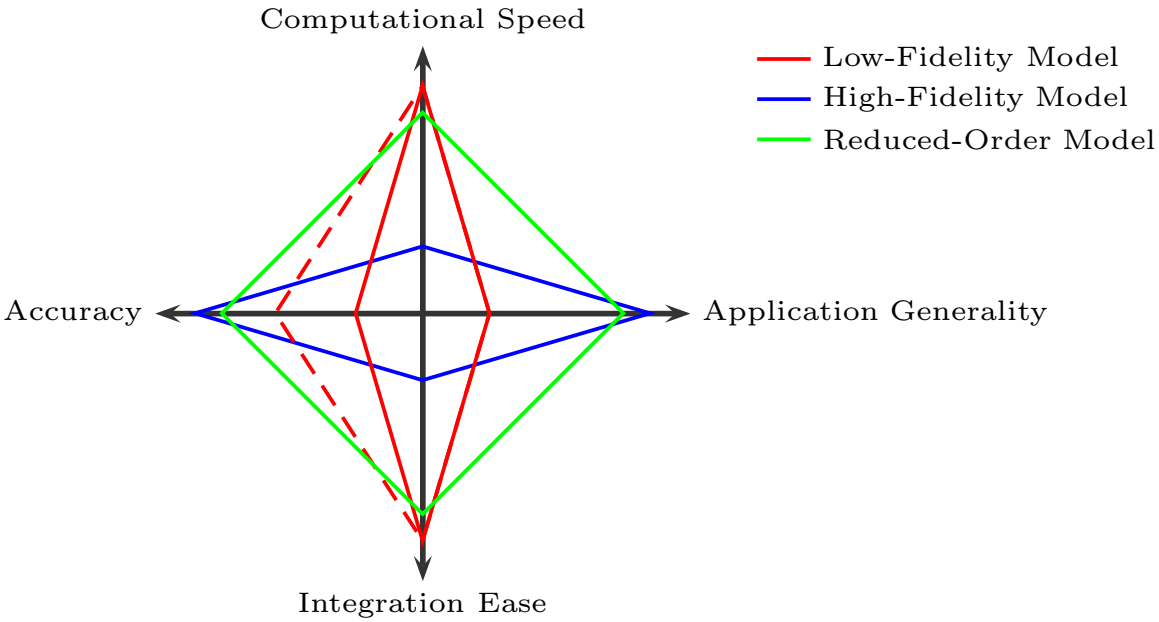


Figure 1: Characteristics of low- and high-fidelity analyses

To this end, researchers have employed surrogate modeling [76] and reduced-order modeling (ROM) [40] in order to reduce a high-fidelity analysis to its tractable substitute. When one deals with a large number of input parameters, typically less than 20, for a few or a small number of output responses, a surrogate modeling approach is suitable. Examples of various surrogate modeling techniques are the response surface methodology (RSM), kriging, neural networks, radial basis functions (RBFs), and support vector machines (SVMs). In contrast, when one has to manage relatively huge information from high-fidelity simulations for a small number of input parameters, an ROM approach is preferable. In computational physics, the following techniques are usually employed for ROM: Volterra series representations, proper orthogonal decomposition (POD), harmonic balance, reduced basis methods, and Krylov subspace methods. Of the two approaches, i.e., surrogate modeling and ROM,

the size of output responses that must be held determines a proper simplifying scheme. For example, surrogate modeling is a better choice for emulating the variations of an aircraft performance metric with changes in airframe geometry parameters[49] whereas ROM is convenient for maintaining an entire airfoil surface pressure distribution for the purpose of airfoil shape design.[36]

1.1.1 Proper Orthogonal Decomposition (POD)

High-fidelity aerodynamic analyses have widely utilized POD, also known as principal component analysis (PCA)[25,74] or the Karhunen-Loève transform (KLT). Since POD is capable of revealing conspicuous features from observations, its applications are quite diverse across numerous engineering realms. For instance, in the field of image processing and pattern recognition, McGregor et al.[50] showed that a POD basis obtained from numerical simulations such as computational fluid dynamics (CFD) can enhance poor blood flow images contaminated with noise. Likewise, in turbulent flow analysis, POD has been adopted to identify the distinctive characteristics of turbulence flow.[18]

Theoretically, POD extracts an empirical basis from observations such that the obtained linear basis describes the maximum variability of observations. Depending on various applications, a POD basis goes by different names such as principal axes, principal components, empirical modes, eigenfunctions, and eigenfaces. Such a broad usage of POD in distinctive application contexts stems from its desirable properties:[63] specifically, (i) An orthogonal basis obtained by POD is optimal for subspace projection in terms of a mean squared error, and (ii) data compression and reconstruction are easily achievable through simple linear transformation. In formulation, POD has two variants: the original POD and the method of snapshots by Sirovich.[77]

Given a POD basis, POD-based ROM reduces the dimensionality of high-dimensional data by projecting them onto a low-dimensional subspace spanned by a POD basis. For example, turbulence flow analysis with direct numerical simulation (DNS) has employed POD to create the low-dimensional dynamic model of turbulence.[7] Similarly, multidisciplinary analyses such as aeroelasticity[80] and flow control[79] relied on POD-base ROM to simplify

unsteady CFD analysis. For both turbulent and unsteady flow analyses, a POD basis is conducive to transforming flow-governing equations, time- and space-dependent partial differential equations (PDEs), into those in forms of only time-dependent ordinary differential equations (ODEs) that are easier to solve. In addition to the single time-parametric POD-based ROM applications, the POD-based ROM scheme adopted interpolation[43], and its applications were extended to design studies that dealt with relatively few design variables.[5,36,37] Since the prediction accuracy of POD-based ROM hinges on the quality of modal coefficients weighting basis functions, several researchers have investigated various multivariate data interpolation techniques[2] for proper modal coefficient estimation.

For instance, Bui-Thanh, Damodaran, and Willcox[5,6] utilized a cubic spline interpolation for airfoil shape design with two design variables using the POD-based ROM of steady CFD analysis. Similarly, Mifsud and his colleagues[52,53] employed RSM to pseudo-continuously represent modal coefficients for three input parameters in weapon aerodynamics studies. Furthermore, Lee et al.[34] capitalized on neural networks to estimate modal coefficients for the ROM of the numerical propulsion system simulation (NPSS), dealing with six engine parameters. Last, POD can also benefit stochastic computational aerodynamics in conjunction with polynomial chaos for efficient probabilistic uncertainty propagation. In particular, Acharjee and Zabaras[1] showed that POD and polynomial chaos can separately account for the decomposition of spatial and random domains, respectively.

In a geometric sense, POD generates an orthonormal basis by rotating a current coordinate system upon which high-dimensional data are recorded. This concept of basis change through axis rotation is familiar in the forms of principal stresses and principal bending axes in structural analysis. For example, Mohr's circle, which is used to evaluate principal stresses, share the same idea of axis rotation as POD. Figure 2 illustrates that one can annihilate shear stresses by rotating current axes to principal axes on which principal stresses remain as the only stress components, as shown in Eq. (1).

$$\begin{bmatrix} \sigma_x & \tau_{xy} \\ \tau_{xy} & \sigma_y \end{bmatrix} \quad \Longrightarrow \quad \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \tag{1}$$

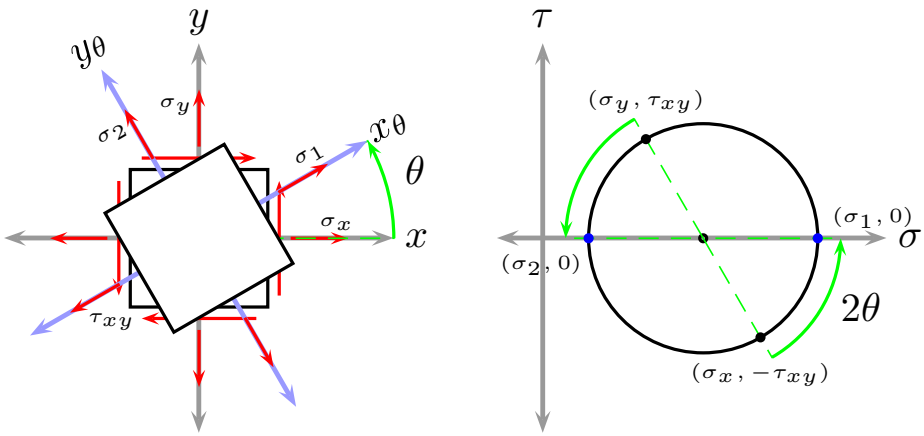


Figure 2: Mohr's circle

The other familiar example is principal bending axes in three-dimensional Euler-Bernoulli beam theory.[3] Principal bending axes eliminate cross-bending stiffness by decoupling two bending moment equations. Along with shifting the origin of principal bending axes to a centroid, as depicted in Figure 3, the centroidal principal axes of bending can eliminate all off-diagonal terms in a sectional stiffness matrix, as shown in Eq. (2). As a result, one can fully decouple structural governing equations for beam analysis, resulting in three decoupled governing equations: one axial and two bending equations.

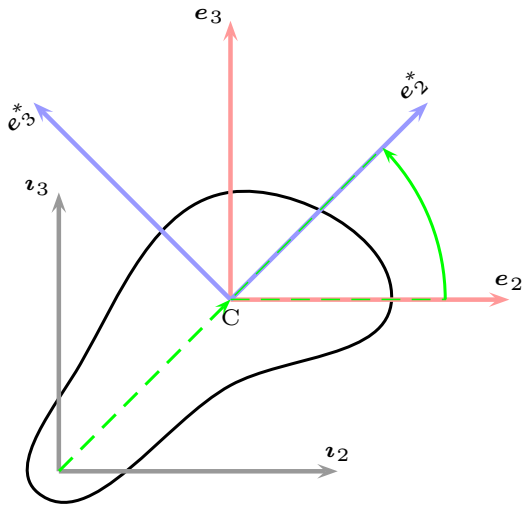


Figure 3: Principal axes of bending

$$\begin{bmatrix} S & S_3 & -S_2 \\ S_3 & H_{22} & -H_{23} \\ -S_2 & H_{23} & H_{33} \end{bmatrix} \quad \Longrightarrow \quad \begin{bmatrix} S^* & 0 & 0 \\ 0 & H_{22}^{c*} & 0 \\ 0 & 0 & H_{33}^{c*} \end{bmatrix} \tag{2}$$

All in all, the central ideas of POD, principal stresses, and principal bending axes are to rotate axes for the diagonalization of a covariance matrix, a stress tensor, and a sectional stiffness tensor, respectively. In matrix theory, all three matrices can be characterized as a symmetric positive definite matrix whose diagonalization relies on the spectral theorem.[9]

Conceptually familiar to POD, POD-based ROM is a form of an approximate solution in structural dynamics. In the Euler-Bernoulli beam theory,[3] the governing PDE of a uniform beam under free vibration is given by

$$H_{33}^{c}\frac{\partial^4 v}{\partial x^4} + m\frac{\partial^2 v}{\partial t^2} = 0, \tag{3}$$

where v is a transverse displacement, H_{33}^{c} is a centroidal bending stiffness, and m is the mass per unit length. The solution of Eq. (3) can be presumed as

$$v(t,x) = \sum_{j=1}^{n} \xi_j(t)\phi_j(x),$$

where ϕ_j is a time-independent mode function and ξ_j is a weighting coefficient corresponding to ϕ_j. For a proper solution, mode functions are required to admit several properties.[17] For example, (i) they satisfy geometric boundary conditions, (ii) they are linearly independent, and so forth. Given mode functions, appropriate modal coefficients can be determined by methods of weighted residuals,[87] namely Galerkin methods, least-squares methods, and others. Analogous to Eq. (3), POD-based ROM has the same form such that

$$\mathbf{y}(\boldsymbol{\vartheta}, \mathbf{x}) = \sum_{j=1}^{n} \alpha_j(\boldsymbol{\vartheta})\boldsymbol{\Phi}_j(\mathbf{x}) + \bar{\mathbf{y}}, \tag{4}$$

where $\boldsymbol{\Phi}_j$ is a POD basis invariant to parameter $\boldsymbol{\vartheta}$, α_j is a coefficient for $\boldsymbol{\Phi}_j$, and $\bar{\mathbf{y}}$ is the sample mean of observations. In Eq. (38), $\bar{\mathbf{y}}$ accounts for a deviation from the origin of a current coordinate system, and $\boldsymbol{\Phi}_j$ weighted by α_j delineates variations in observations. Although structural dynamics analysis can analytically predetermine basis functions to

solve Eq. (3), POD-based ROM necessitates POD to empirically find basis functions from compiled data.

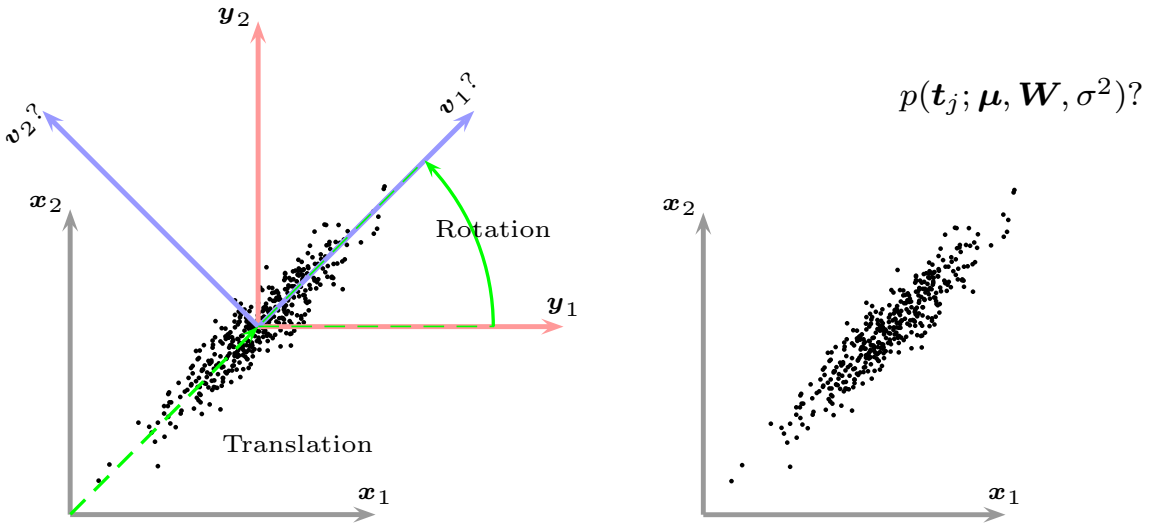
1.1.2 Gappy Proper Orthogonal Decomposition

Since POD is impotent even in the slightest absence of data, Everson and Sirovich[13] devised gappy POD, which solves a least-squares problem defined with a "gappy" norm and a "POD" basis. They proposed the gappy norm as an alternative to an L^2 norm because the L^2 norm fails to deal with incomplete data for an estimation residual evaluation; thus, the gappy norm is simply the L^2 norm that neglects unavailable data. Through a least-squares approach, gappy POD can extract a POD basis from an incomplete data set as well as restore unknown data, treated as missing, in observations. Originally, gappy POD was proposed for the restoration of randomly marred human face images, but its applications have spread to other engineering realms; missing data estimation is so general that it can epitomize diverse problems insofar as appropriate missing data forms can be devised for them.

After Bui-Thanh[4] introduced gappy POD to aerospace engineering, several researchers applied it to not only literally approximating missing data but also virtually addressing various problems seemingly irrelevant to missing data estimation. For instance, Venturi and Karniadakis[84] tested gappy POD with other reconstruction methods such as local kriging and local linear interpolation. Likewise, Murray and Ukeiley[58] and Murray and Seiner[57] utilized gappy POD to repair spurious measurements of particle image velocimetry (PIV) in experimental flow analysis. Meanwhile, other researchers found the potential of gappy POD in the areas outside the context of flow data restoration. For example, Bui-Thanh, Damodaran, and Willcox[5] solved an inverse airfoil design problem by treating unknown airfoil coordinates at desired surface pressures as missing. Moreover, Bui-Thanh[4] used gappy POD for parametric flowfield prediction, as Willcox[86] did for unsteady flow reconstruction and effective sensor placement. For variable fidelity analysis, Robinson et al.[62] capitalized on gappy POD to map discrepancies between low- and high-fidelity analyses due to their differences in resolution.

1.1.3 Probabilistic Principal Component Analysis (PPCA)

Intriguingly, the applications of both POD and gappy POD can be tackled from a probabilistic point of view with probabilistic principal component analysis (PPCA). In order to impart a probability model to PCA, i.e., POD, Tipping and Bishop[82] formulated a probabilistic generalization of PCA, termed PPCA. After Gaussian probabilities are assumed for the variables of the PPCA factor analysis model, PPCA ends up with a Gaussian probability model that delineates given observations. Because of this formulation perspective change from deterministic to probabilistic, the PCA problem of finding an orthogonal basis by axis rotation is recast into a PPCA problem of finding the maximum likelihood estimate (MLE) of PPCA parameters, namely mean $\boldsymbol{\mu}$, factor-loading matrix $\mathbf{W}$, and variance σ^2. As an illustration, Figure 4 depicts dissimilar problem interpretations of both PCA and PPCA for the same observations. In Figure 4, each point of view is associated with a disparate mathematical framework: Figure 4(a) with linear algebra and Figure 4(b) with probability and statistics theories. Since Tipping and Bishop[82] proposed PPCA, it has been utilized for such applications as speech recognition[70] and the determination of empirical orthogonal functions in the presence of missing values for oceanography.[19]



(a) Deterministic (b) Probabilistic

Figure 4: Change of formulation perspectives

For a statistical parameter inference, Tipping and Bishop[82] employed an expectation-maximization (EM) algorithm,[10] an iterative method conducive to the presence of missing or hidden data. With the help of the EM algorithm, they developed an EM algorithm for PPCA (EM-PCA) that alternates between an expectation step (E-step) and a maximization step (M-step) to yield PPCA MLEs through iterations. More importantly, by virtue of the EM algorithm, the EM-PCA can naturally deal with not only a complete but also an incomplete data set; in other words, the EM-PCA is able to address the applications of both POD and gappy POD. For instance, given a complete data set, the EM-PCA can find an orthogonal basis $\mathbf{V}_q$ through its non-orthogonal factor-loading matrix $\mathbf{W}$ after orthogonalization that removes additional rotation $\mathbf{R}$, as shown in Figure 5. Likewise, given an incomplete data set, the EM-PCA can approximate missing, unknown observations through the expectation of an observation probability model.

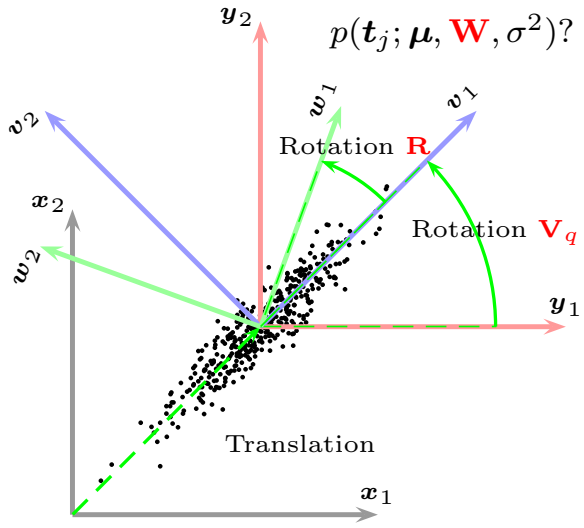


Figure 5: Relationship between $\mathbf{W}$ and $\mathbf{V}$

1.2 Research Questions and Hypotheses

Having established that the EM-PCA can handle the applications of both POD and gappy POD, this research formulates its primary research objectives and methodological hypotheses as follows.

Research Objective 1. To facilitate the use of the EM-PCA in addressing the problems of aerospace engineering, this research attempts to theoretically and numerically compare and

contrast the EM-PCA and to both POD and gappy POD for basis extraction and missing data estimation.

Methodological Hypothesis 1. The EM-PCA yields identical results to those of POD and gappy POD, but it is computationally more efficient than POD and gappy POD in terms of computational time.

To demonstrate the potential of the EM-PCA, this research applies the EM-PCA to the reduced-order modeling of NPSS as an efficient method for basis extraction. Furthermore, since the EM-PCA is capable of dealing with incomplete data, it is utilized for PIV data restoration as an efficient method for missing data estimation. In pursuing Research Objective 1, this research attempts to address the following research questions and corresponding hypotheses to separately evaluate Methodological Hypothesis 1 for different types of data sets.

Research Question 1.1. For an intact data set, is the EM-PCA computationally competitive with POD methods for basis extraction?

Hypothesis 1.1. For an intact data set, the EM-PCA takes less computational time than POD.

Research Question 1.2. For an incomplete data set whose missing data are only at a single snapshot, is the EM-PCA computationally competitive with gappy POD for basis extraction and missing data estimation?

Hypothesis 1.2. For an incomplete data set whose missing data are only at a single snapshot, the EM-PCA takes less computational time than gappy POD.

Research Question 1.3. For an incomplete data set whose missing data are across all the snapshots, is the EM-PCA computationally competitive with gappy POD for basis extraction and missing data estimation?

Hypothesis 1.3. For an incomplete data set whose missing data are across all the snapshots, the EM-PCA takes less computational time than gappy POD.

Research Question 1.1 aims to test Hypothesis 1.1 for a complete data set, and both Research Questions 1.2 and 1.3 aim to test Hypotheses 1.2 and 1.3 for an incomplete data set. Note that an incomplete data set is further decomposed into two missing data types based on its missing data characteristic observed in the literature of gappy POD applications; one is deficient of data only at a single snapshot, such as flow data assimilation[84] or inverse airfoil design,[5] and the other is absent of data across all the snapshots, such as PIV data restoration.[57,58] To verify the aforementioned hypotheses, this research designs several numerical experiments for the comparison of the EM-PCA to POD and to gappy POD as follows:

Experiments

- Complete data sets generated by full potential equation (FPE) and Euler CFD solvers
- Incomplete data sets generated by an Euler CFD solver after 30% of simulation data are artificially removed

Throughout the comparative studies in this research, the computational performance of the algorithms are measured with two metrics, namely computational time and the number of iterations; the former indicates an overall performance assessment, and the latter reveals an estimation error reduction per iteration.

As this research strives to answer Research Questions 1.2 and 1.3, it found no research literature that elucidates the theoretical relationship between the EM-PCA and gappy POD. Hence, in an effort to facilitate the comparative studies of the EM-PCA and gappy POD, the second research objective arises.

Research Objective 2. To compare and contrast the EM-PCA to gappy POD, this research attempts to identify the formulation similarities and disparities of the EM-PCA and gappy POD.

Due to insufficient knowledge regarding the relationship between the EM-PCA and gappy POD, this research cannot further develop a methodological hypothesis relevant to Research Objective 2 and subsequent research questions. However, this research will address Research Objective 2 in detail with the help of a unifying least-squares perspective later

in Chapter 4 as it accumulates observations related to their formulation similarities and disparities.

1.3 Contributions and Dissertation Outline

In summary, the objective of this research is to promote the EM-PCA in lieu of POD and gappy POD to effectively address basis extraction and missing data estimation applications in aerospace engineering. Overall, the main contributions of this dissertation are as follows: (i) It introduces the EM-PCA in the realm of aerospace engineering; (ii) it compares the EM-PCA with both POD and gappy POD quantitatively and qualitatively, specifically through a comparative study of the EM-PCA and gappy POD that (ii–1) provides a unifying least-squares perspective that integrates both the EM-PCA and gappy POD within a common formulation framework., (ii–2) identifies the similarities and disparities of the EM-PCA and gappy POD, and (ii–3) quantifies the theoretical and numerical effects of different bases and norms of the EM-PCA and gappy POD; and (iii) it demonstrates the benefits of the EM-PCA over gappy POD in several aerospace applications: (iii–1) the ROM of NPSS to facilitate airframe- and engine-integrated aircraft design, and (iii–2) the restoration of PIV data to efficiently rectify spurious PIV measurements.

Overall, this thesis is organized as follows. After the introduction, Chapter 2 articulates the theories of deterministic and probabilistic POD formulations; the former comprises POD and gappy POD, and the latter includes PPCA along with the EM-PCA. Subsequently, Chapters 3 and 4 present two comparative studies that theoretically and numerically examine the EM-PCA to compare it to both POD and gappy POD. In particular, in the second comparative study, Chapter 4 analyzes the dissimilar formulations of gappy POD and the EM-PCA from the unifying least-squares perspective. Moreover, it formulates the hybrid algorithms of gappy POD and the EM-PCA so as to delve into the ramifications of their disparate bases and norms for missing data estimation. Afterwards, to demonstrate the advantages of the EM-PCA, Chapters 5 and 6 illustrate two aerospace applications: POD-based ROM construction for NPSS and PIV data reconstruction as examples of basis

extraction and missing data estimation, respectively. Finally, Chapter 7 closes this dissertation with conclusions and recommendations for future work, followed by Appendices A and B, each of which provides a mathematical proof related to the derivation of the EM-PCA and supplementary results left out of Chapter 5, respectively.

CHAPTER II

THEORY

2.1 Deterministic Formulations

This section briefly describes the two POD formulations, i.e., the original POD and the method of snapshots, to extract an orthogonal basis from a snapshot ensemble of complete data. For convenience, the former and the latter are hereafter denoted as standard POD and snapshot POD, respectively. Subsequently, this section elucidates the formulation of gappy POD, which is extended based on POD, to handle a snapshot ensemble of incomplete data for orthogonal basis extraction as well as missing data estimation.

In order to simplify formulation derivations, this section introduces mean-subtracted quantities, $\dot{\mathbf{y}}_j \in \mathbb{R}^d$ and $\dot{\mathbf{Y}} = \{\dot{\mathbf{y}}_j\}_{j=1}^N \in \mathbb{R}^{d\times N}$, for each snapshot $\mathbf{y}_j \in \mathbb{R}^d$ and snapshot ensemble $\mathbf{Y} = \{\mathbf{y}_j\}_{j=1}^N \in \mathbb{R}^{d\times N}$, respectively. A mean-centered snapshot $\dot{\mathbf{y}}_j$ is evaluated from a snapshot $\mathbf{y}_j$ such that $\dot{\mathbf{y}}_j = \mathbf{y}_j - \bar{\mathbf{y}}$, where $\bar{\mathbf{y}} \in \mathbb{R}^d$ is a sample mean normally determined by $\bar{\mathbf{y}} = (1/N)\sum_{j=1}^N \mathbf{y}_j$. In case of the presence of unknown missing data, the i^{th} element of $\bar{\mathbf{y}}$ is evaluated with only known data such that

$$\bar{y}_i = \frac{1}{\sum_{j=1}^N n_{ij}} \sum_{j=1}^N n_{ij} y_{ij}$$

where $n_{ij} = 1$ if y_{ij} is available; if not, $n_{ij} = 0$. In the same manner, the ensemble of mean-centered snapshots $\dot{\mathbf{Y}}$ is computed from the collection of snapshots $\mathbf{Y}$ such that $\dot{\mathbf{Y}} = \mathbf{Y} - \bar{\mathbf{y}}\mathbf{1}_N^{\mathrm{T}}$, where $\mathbf{1}_N \in \mathbb{R}^N$ is a column vector of N ones given by $\mathbf{1}_N = (1,\ldots,1)^{\mathrm{T}}$. Without loss of generality, this section will denote $\dot{\mathbf{y}}_j$ and $\dot{\mathbf{Y}}$ as $\mathbf{y}_j$ and $\mathbf{Y}$, respectively, for notational convenience.

2.1.1 Proper Orthogonal Decomposition

In view of a linear algebra framework, POD pertains to changing a given basis into an orthogonal basis, i.e., principal components, through diagonalizing a sample covariance matrix of observations.[74] In formulation, POD has two versions; standard POD deals with a

d-by-d sample covariance matrix of the column vectors of $\mathbf{Y}$, and so does snapshot POD with an N-by-N sample covariance matrix of the row vectors of $\mathbf{Y}$. Depending on the size of $\mathbf{Y}$, the standard POD method is recommended when $d < N$; otherwise, the snapshot POD method is preferable to the standard POD method for computational efficiency. For example, the snapshot POD method is a general choice for high-fidelity aerodynamic simulation data because the number of grid points d is usually much larger than the number of snapshots N. Note that the rank of a snapshot ensemble $\text{rank}(\mathbf{Y})$ is typically less than either d or N such that $\text{rank}(\mathbf{Y}) < \min\{d, N\}$. Although there is a continuous POD formulation, this section delineates only discrete POD formulations that are useful for a snapshot ensemble obtained from successive physical experiments or numerical simulations; for the other continuous formulations, refer to the work of Sirovich.[77]

2.1.1.1 Standard Method

For a given snapshot ensemble $\mathbf{Y}$, the standard POD method extracts a d number of orthogonal basis vectors spanning the column space of $\mathbf{Y}$. A sample covariance matrix of column vectors $\mathbf{S} \in \mathbb{R}^{d \times d}$ is evaluated by

$$\mathbf{S} = \frac{1}{N-1}\mathbf{Y}\mathbf{Y}^{\mathrm{T}},$$

and the orthogonal basis, on which $\mathbf{S}$ is uncorrelated, can be found with either eigenvalue decomposition (EVD) or singular value decomposition (SVD) of $\mathbf{S}$ such that

$$\mathbf{S}\mathbf{V} = \mathbf{V}\mathbf{\Lambda} \quad \text{or} \quad \mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{\mathrm{T}},$$

which yields eigenvectors $\mathbf{V} \in \mathbb{R}^{d \times d}$ that span the column space of $\mathbf{Y}$ and a diagonal matrix $\mathbf{\Lambda} \in \mathbb{R}^{d \times d}$ that lists a d number of eigenvalues corresponding to $\mathbf{V}$ in its diagonal. Because $\mathbf{S}$ is a symmetric positive semi-definite matrix, eigenvectors in $\mathbf{V}$ are orthogonal to each other, and eigenvalues in $\mathbf{\Lambda}$ are real values greater than or equal to zero.

2.1.1.2 Snapshot Method

For efficient basis extraction, Sirovich[77] devised the method of snapshots, the snapshot POD method, to obviate handling $\mathbf{S}$ whose size is prone to large for $\mathbf{Y}$ whose $d > N$. The

snapshot POD method evaluates a sample covariance matrix $\mathbf{R} \in \mathbb{R}^{N \times N}$ of the row space of $\mathbf{Y}$ such that

$$\mathbf{R} = \frac{1}{d-1}\mathbf{Y}^{\mathrm{T}}\mathbf{Y},$$

and applies EVD or SVD to $\mathbf{R}$ as follows:

$$\mathbf{R}\mathbf{U} = \mathbf{U}\mathbf{\Lambda}_N \quad \text{or} \quad \mathbf{R} = \mathbf{U}\mathbf{\Lambda}_N\mathbf{U}^{\mathrm{T}},$$

where an eigenvector matrix $\mathbf{U} \in \mathbb{R}^{N \times N}$ forms the orthogonal basis of the row space of $\mathbf{Y}$, and a diagonal matrix $\mathbf{\Lambda}_N \in \mathbb{R}^{N \times N}$ contains an N number of corresponding eigenvalues of $\mathbf{U}$ in diagonal. Analogous to $\mathbf{S}$, $\mathbf{R}$ is a symmetric positive semi-definite matrix, thus $\mathbf{U}$ is orthogonal and $\mathbf{\Lambda}$ is semi-positive real. Note that the above described procedures are identical to invoking the previous standard POD method to a transposed snapshot ensemble $\mathbf{Y}^{\mathrm{T}}$. Once the row space basis $\mathbf{U}$ is achieved, it is required to be transformed into the column space basis such that

$$\mathbf{V}_N = \mathbf{Y}\mathbf{U}$$

with the use of $\mathbf{Y}$ as a linear transformation such that $\mathbf{Y} : \mathbb{R}^N \mapsto \mathbb{R}^d$. Finally, unlike the standard POD method producing a d number of basis vectors $\mathbf{V}$, the snapshot POD method generates an N number of basis vectors $\mathbf{V}_N \in \mathbb{R}^{d \times N}$ spanning the column space of $\mathbf{Y}$. As long as N is large enough to rank($\mathbf{Y}$), the snapshot POD is sufficient to extract the essential orthogonal basis vectors of $\mathbf{Y}$.

2.1.1.3 POD-Based Approximation

Provided that orthogonal basis vectors $\mathbf{V}$ is invariant for the given $\mathbf{Y}$ and $q < \mathrm{rank}(\mathbf{Y})$, $\mathbf{Y}$ can be approximated as a linear combination of the first q dominant basis vectors of $\mathbf{V}$ such that

$$\mathbf{Y} \approx \mathbf{V}_q\mathbf{A},$$

where $\mathbf{A}$ is a coefficient matrix. For the best approximation of $\mathbf{Y}$ with $\mathbf{V}_q$, $\mathbf{A}$ can be found as the solution of a least-squares problem:

$$\text{min.} \quad \|\mathbf{Y} - \mathbf{V}_q\mathbf{A}\|_{L^2}^2 \quad \text{w.r.t.} \quad \mathbf{A},$$

which determines $\mathbf{A}$ as follows:

$$\mathbf{A} = \left(\mathbf{V}_q^{\mathrm{T}}\mathbf{V}_q\right)^{-1}\mathbf{V}_q^{\mathrm{T}}\mathbf{Y}.$$

2.1.2 Gappy Proper Orthogonal Decomposition

2.1.2.1 Formulation

In case of without any missing data, an arbitrary snapshot $\mathbf{y}_j$ that belongs to a snapshot ensemble $\mathbf{Y}$ can be represented as a linear combination of a POD basis $\mathbf{V}_q$ such that $\mathbf{y}_j \approx \widetilde{\mathbf{y}}_j = \mathbf{V}_q\mathbf{b}_j$ as shown in Section 2.1.1.3. For the best approximation, a modal coefficient $\mathbf{b}_j \in \mathbb{R}^q$ is evaluated such that it minimizes a squared error

$$\text{min.} \quad \left\|\mathbf{y}_j - \mathbf{V}_q\mathbf{b}_j\right\|_{L^2}^2 \quad \text{w.r.t.} \quad \mathbf{b}_j, \tag{5}$$

which yields the optimal coefficient $\mathbf{b}_j = (\mathbf{V}_q^{\mathrm{T}}\mathbf{V}_q)^{-1}\mathbf{V}_q^{\mathrm{T}}\mathbf{y}_j$ for the given basis $\mathbf{V}_q$.

Similarly in the presence of gappy data, missing data elements in an incomplete snapshot $\mathring{\mathbf{y}}_j$ can be restored by adopting the same least-squares approach in Eq. (5) using $\mathring{\mathbf{y}}_j \approx \mathbf{V}_q\mathbf{b}_j$ provided that a POD basis $\mathbf{V}_q$ is available.

$$\text{min.} \quad \left\|\mathring{\mathbf{y}}_j - \mathbf{V}_q\mathbf{b}_j\right\|_{L^2}^2 \quad \text{w.r.t.} \quad \mathbf{b}_j. \tag{6}$$

However, the unknown missing elements in $\mathring{\mathbf{y}}_j$ causes to fail the evaluation of the above squared residual expressed in the L^2 norm in Eq. (6). In order to work around this issue, Everson & Sirovich[13] came up with the gappy norm defined with the gappy inner product $(\,\cdot\,,\,\cdot\,)_n$ on $\mathbb{R}^d$ such that

$$\|\mathbf{y}_j\|_n^2 := (\mathbf{y}_j, \mathbf{y}_j)_n = (\mathbf{n}_j \circ \mathring{\mathbf{y}}_j, \mathbf{n}_j \circ \mathring{\mathbf{y}}_j)_{L^2} = \|\mathbf{n}_j \circ \mathring{\mathbf{y}}_j\|_{L^2}^2, \tag{7}$$

where $\circ$ denotes a Hadamard product, i.e., point-wise multiplication, and $\mathbf{n}_j$ is a mask vector corresponding to an incomplete snapshot $\mathring{\mathbf{y}}_j$ to screen out missing data in $\mathring{\mathbf{y}}_j$. The mask vector $\mathbf{n}_j$ is defined by

$$n_{ij} = \begin{cases} 0 & \text{if } y_{ij} \text{ is missing,} \\ 1 & \text{if } y_{ij} \text{ is known.} \end{cases} \quad \text{for} \quad i = 1, \ldots, d. \tag{8}$$

Note that masking by $\mathbf{n}_j$ for $\mathring{\mathbf{y}}_j$ implies to assign a sample mean to each missing data element in $\mathring{\mathbf{y}}_j$ since a snapshot is beforehand treated as mean-centered for convenience in Section 2.1. With the help of the gappy norm, the previous squared residual in Eq. (6) is rephrased as

$$r_j^2 = \left\|\mathring{\mathbf{y}}_j - \mathbf{V}_q\mathbf{b}_j\right\|_{L^2}^2 = \left\|\mathbf{n}_j \circ (\mathring{\mathbf{y}}_j - \mathbf{V}_q\mathbf{b}_j)\right\|_{L^2}^2 = \left\|\mathbf{y}_j - \mathbf{V}_q\mathbf{b}_j\right\|_n^2, \tag{9}$$

and the least-squares problem for the gappy POD is

$$\text{min.} \quad \left\|\mathbf{y}_j - \mathbf{V}_q\mathbf{b}_j\right\|_n^2 \quad \text{w.r.t.} \quad \mathbf{b}_j. \tag{10}$$

In order to find $\mathbf{b}_j$ satisfying Eq. (10), one can rephrase $\mathbf{V}_q\mathbf{b}_j$ in a vector form such that $\mathbf{V}_q\mathbf{b}_j = \sum_{i=1}^{q} b_{ij}\mathbf{v}_i$ and evaluates the first derivative of r_j^2 with respect to $\mathbf{b}_j$:

$$\frac{\partial}{\partial \mathbf{b}_j}\left\|\mathbf{y}_j - \mathbf{V}_q\mathbf{b}_j\right\|_n^2 = \frac{\partial}{\partial \mathbf{b}_j}\left\|\mathbf{n}_j \circ (\mathbf{y}_j - \mathbf{V}_q\mathbf{b}_j)\right\|_{L^2}^2 = \frac{\partial}{\partial \mathbf{b}_j}\left\|\mathbf{n}_j \circ (\mathbf{y}_j - \sum_{i=1}^{q} b_{ij}\mathbf{v}_i)\right\|_{L^2}^2.$$

In detail, the evaluation of the first-order partial derivative with respect to $\mathbf{b}_j$ can be expanded as follows:

$$\begin{aligned}
&= \frac{\partial}{\partial \mathbf{b}_j}\left\|(\mathbf{n}_j \circ \mathbf{y}_j) - \sum_{i=1}^{q} b_{ij}(\mathbf{n}_j \circ \mathbf{v}_i)\right\|_{L^2}^2 \\
&= \frac{\partial}{\partial \mathbf{b}_j}\left((\mathbf{n}_j \circ \mathbf{y}_j) - \sum_{i=1}^{q} b_{ij}(\mathbf{n}_j \circ \mathbf{v}_i)\right)^{\mathrm{T}}\left((\mathbf{n}_j \circ \mathbf{y}_j) - \sum_{i=1}^{q} b_{ij}(\mathbf{n}_j \circ \mathbf{v}_i)\right) \\
&= \frac{\partial}{\partial \mathbf{b}_j}\left((\mathbf{n}_j \circ \mathbf{y}_j)^{\mathrm{T}}(\mathbf{n}_j \circ \mathbf{y}_j) - 2\sum_{i=1}^{q} b_{ij}(\mathbf{n}_j \circ \mathbf{y}_j)^{\mathrm{T}}(\mathbf{n}_j \circ \mathbf{v}_i)\right. \\
&\quad \left. + \sum_{i=1}^{q}\sum_{k=1}^{q} b_{ij}b_{kj}(\mathbf{n}_j \circ \mathbf{v}_i)^{\mathrm{T}}(\mathbf{n}_j \circ \mathbf{v}_k)\right) \\
&= -2(\mathbf{n}_j \circ \mathbf{y}_j)^{\mathrm{T}}(\mathbf{n}_j \circ \mathbf{v}_i) + 2\sum_{k=1}^{q} b_{kj}(\mathbf{n}_j \circ \mathbf{v}_i)^{\mathrm{T}}(\mathbf{n}_j \circ \mathbf{v}_k).
\end{aligned}$$

Finally, the optimal b_{ij} is found such that

$$b_{ij} = \left(\sum_{k=1}^{q}(\mathbf{n}_j \circ \mathbf{v}_i)^{\mathrm{T}}(\mathbf{n}_j \circ \mathbf{v}_k)\right)^{-1}(\mathbf{n}_j \circ \mathbf{y}_j)^{\mathrm{T}}(\mathbf{n}_j \circ \mathbf{v}_i), \tag{11}$$

and in a matrix form, it is evaluated as

$$\begin{bmatrix} \\ \\ b_{ij} \\ \\ \\ \end{bmatrix} = \begin{bmatrix} \\ \\ — \quad (\mathbf{n}_j \circ \mathbf{v}_i)^{\mathrm{T}}(\mathbf{n}_j \circ \mathbf{v}_k) \quad — \\ \\ \\ \end{bmatrix}^{-1} \begin{bmatrix} \\ \\ (\mathbf{n}_j \circ \mathbf{y}_j)^{\mathrm{T}}(\mathbf{n}_j \circ \mathbf{v}_i) \\ \\ \\ \end{bmatrix}$$

for $k = 1 \sim q$. After all, the least-squares problem of gappy POD in Eq. (10) is reduced to a system of q linear equations such that

$$M_i b_{ij} = f_{ij}, \quad \text{where} \quad M_i = \sum_{k=1}^{q} (\mathbf{n}_j \circ \mathbf{v}_i)^{\mathrm{T}}(\mathbf{n}_j \circ \mathbf{v}_k), \quad \text{and} \quad f_{ij} = (\mathbf{n}_j \circ \mathbf{y}_j)^{\mathrm{T}}(\mathbf{n}_j \circ \mathbf{v}_i),$$

for $i = 1, \ldots, q$. Note that the optimal b_{ij} of gappy POD in Eq. (11) has an identical form to the ordinary least-squares coefficient b_{ij} in Eq. (5), determined by

$$b_{ij} = \left(\sum_{k=1}^{q} (\mathbf{v}_i^{\mathrm{T}} \mathbf{v}_k) \right)^{-1} (\mathbf{y}_j^{\mathrm{T}} \mathbf{v}_i),$$

except that every vector in Eq. (11) is masked by $\mathbf{n}_j$. Once $\mathbf{b}_j$ is obtained from Eq. (11), the missing data of $\mathring{\mathbf{y}}_j$ can be estimated through a linear combination of $\mathbf{V}_q$ and $\mathbf{b}_j$ such that

$$\mathring{y}_{ij} = \begin{cases} (\mathbf{V}_q \mathbf{b}_j)_{ij} & \text{if } n_{ij} = 0, \\ y_{ij} & \text{if } n_{ij} = 1, \end{cases} \quad \text{for} \quad i = 1, \ldots, d,$$

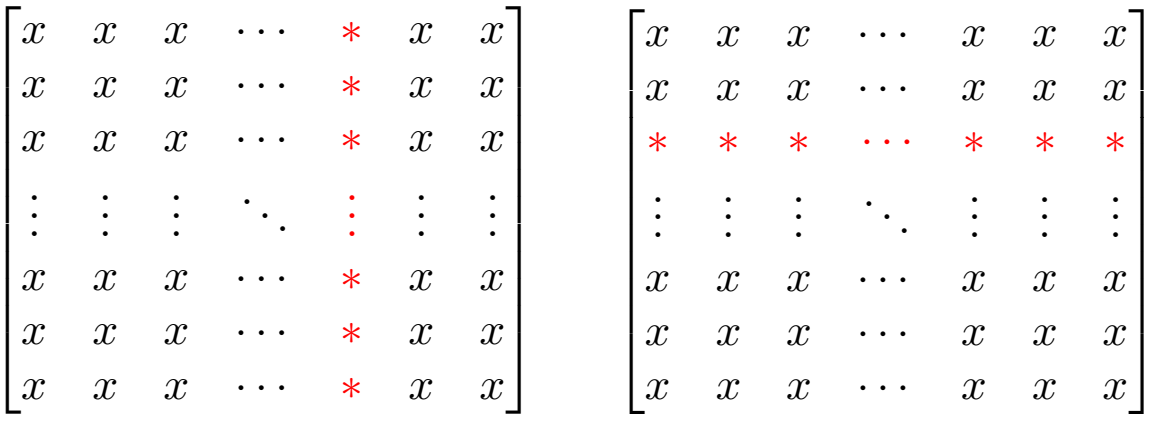
leaving known data intact. Note that the gappy POD presumes a POD basis $\mathbf{V}_q$ is obtainable, and yet the true $\mathbf{V}_q$ is not known a priori unless all the data of snapshots are available. Therefore, gappy POD has to repeat basis and coefficient evaluations through iterations; the former derives an estimated basis $\widetilde{\mathbf{V}}_q$ from an intermediate snapshot ensemble $\widetilde{\mathbf{Y}}$, and the latter rectifies the estimated snapshot ensemble $\widetilde{\mathbf{Y}}$ by repairing missing data using the previous basis estimate $\widetilde{\mathbf{V}}_q$.

2.1.2.2 Limitations of Gappy POD

Although the gappy POD formulation in Section 2.1.2.1 is designed for missing data estimation, gappy POD is not a panacea that can treat all types of missing data; it is inapplicable

to two particular structures of missing data that miss either an entire row or an entire column of a data set. For illustration, Figure 6 delineates the two unusual types of missing data, which break down gappy POD. One of the extreme cases, shown in Figure 6(a), is completely absent of one of snapshots, which causes a null mask vector, i.e., $\mathbf{n}_j = 0$, for the corresponding entirely missing snapshot $\mathring{\mathbf{y}}_j$. Because of this empty mask vector, the least-squares coefficient $\mathbf{b}_j$ in Eq. (11) vanishes; consequently, gappy POD is unable to estimate all missing data of the relevant missing snapshot.

Similarly, Figure 6(b) delineates the other missing data type, whose total observations at a certain measurement point are missing, indicating that an entire row of a snapshot ensemble is unavailable. This case prevents gappy POD from computing a sample mean at a specific location, and thus, it cannot properly initialize missing data at the location with a sample mean. Note that the two extreme missing data cases in Figure 6, both of which fail gappy POD, are impractical other than a formulation aspect. For instance, one can easily remove an entire data-missing snapshot from a snapshot ensemble circumventing the first missing data type in Figure 6(a). Likewise, one would avert the second missing data type in Figure 6(b), caused by inherent measurement limitations, if all the deficient observations matter for a physical or numerical experimental purpose.



(a) An entire snapshot missing

(b) All observations missing at a measurement location

Figure 6: Two extreme missing data cases*

*A $*$ denotes a missing data element.

2.2 Probabilistic Formulation

This section presents a probabilistic interpretation of the standard POD, namely PPCA. For a statistical parameter interference, PPCA invokes the EM algorithm, resulting in the EM-PCA. By virtue of the EM algorithm, the EM-PCA can also approximate missing data like gappy POD. For the sake of derivation simplicity, as similar to the POD derivations in Section 2.1, both a snapshot $\mathbf{y}_j$ and a snapshot ensemble $\mathbf{Y}$ are treated as mean-centered since $\boldsymbol{\mu}$ is a nuisance parameter[†] in PPCA parameter estimation.

2.2.1 Probabilistic Principal Component Analysis

2.2.1.1 Latent Variable Model

Tipping and Bishop[82] developed PPCA presuming a latent variable model that relates an observed variable $\mathbf{y}_j \in \mathbb{R}^d$ to a linear latent variable[‡] $\mathbf{x}_j \in \mathbb{R}^q$ where $d \gg q$. For an arbitrary observation $\mathbf{y}_j$, the linear latent variable model is given by

$$\mathbf{y}_j(\mathbf{x}_j; \mathbf{W}) = \mathbf{W}\mathbf{x}_j,$$

where $\mathbf{W} \in \mathbb{R}^{d \times q}$ is a factor-loading matrix that represents a linear mapping, i.e., $\mathbf{W} : \mathbf{x}_j \mapsto \mathbf{y}_j$. With the assumption of an additional error $\boldsymbol{\epsilon} \in \mathbb{R}^d$ for $\mathbf{y}_j$ independent of the latent variable $\mathbf{x}_j$, an error accounted observation $\mathbf{t}_j \in \mathbb{R}^d$ is

$$\mathbf{t}_j(\mathbf{x}_j; \mathbf{W}, \boldsymbol{\epsilon}) = \mathbf{W}\mathbf{x}_j + \boldsymbol{\epsilon}. \tag{12}$$

Note that the latent variable model in Eq. (12) conveys the idea of dimensionality reduction because a high-dimensional observation $\mathbf{t}_j$ can be delineated by a low-dimensional latent variable $\mathbf{x}_j$ through the mapping $\mathbf{W}$. For the derivation of the probability density model of $\mathbf{t}_j$, several assumptions are introduced to Eq. (12): (i) a unit isotropic Gaussian distribution for $\mathbf{x}_j$ such that $\mathbf{x}_j \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and (ii) an isotropic Gaussian noise for $\boldsymbol{\epsilon}$ such that $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$. Finally, $\mathbf{t}_j$ ends up with a Gaussian probability model such that $\mathbf{t}_j \sim \mathcal{N}(\mathbf{0}, \mathbf{W}\mathbf{W}^{\mathrm{T}} + \sigma^2\mathbf{I})$.

[†] $\boldsymbol{\mu}$ can be simply evaluated by a sample mean.

[‡] In statistics, a latent variable is a hidden variable that lurks under an observed variable, so it can be inferred only from the observed variable

2.2.1.2 Probability Model

The probability distribution of $\mathbf{t}_j$ given $\mathbf{x}_j$ can be formulated with the help of the probability model of $\boldsymbol{\epsilon}$ given by

$$p(\boldsymbol{\epsilon}; \sigma^2) = (2\pi\sigma^2)^{-d/2} \exp\left(-\frac{1}{2\sigma^2}\boldsymbol{\epsilon}^{\mathrm{T}}\boldsymbol{\epsilon}\right).$$

Since $\boldsymbol{\epsilon} = \mathbf{t}_j - \mathbf{W}\mathbf{x}_j$, the conditional probability of $\mathbf{t}_j$ given $\mathbf{x}_j$ is found as

$$p(\mathbf{t}_j|\mathbf{x}_j; \mathbf{W}, \sigma^2) = (2\pi\sigma^2)^{-d/2} \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{t}_j - \mathbf{W}\mathbf{x}_j\|^2\right)$$

from $p(\boldsymbol{\epsilon})$. With the assumed prior probability of $\mathbf{x}_j$ such that

$$p(\mathbf{x}_j) = (2\pi)^{-q/2} \exp\left(-\frac{1}{2}\mathbf{x}_j^{\mathrm{T}}\mathbf{x}_j\right),$$

the marginal probability of $\mathbf{t}_j$ is

$$p(\mathbf{t}_j; \mathbf{W}, \sigma^2) = \int p(\mathbf{t}_j|\mathbf{x}_j)p(\mathbf{x}_j)d\mathbf{x}_j = (2\pi)^{-d/2}|\mathbf{C}|^{-1/2} \exp\left(-\frac{1}{2}\mathbf{t}_j^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{t}_j\right),$$

where the model covariance $\mathbf{C}$ is defined as

$$\mathbf{C} = \mathbf{W}\mathbf{W}^{\mathrm{T}} + \sigma^2\mathbf{I}.$$

In addition, the posterior probability of $\mathbf{x}_j$ given $\mathbf{t}_j$ is found by the Bayes' rule such that

$$\begin{aligned} p(\mathbf{x}_j|\mathbf{t}_j) &= \frac{p(\mathbf{t}_j|\mathbf{x}_j)p(\mathbf{x}_j)}{p(\mathbf{t}_j)} \\ &= (2\pi)^{-q/2}|\sigma^{-2}\mathbf{M}|^{1/2} \exp\left(-\frac{1}{2}\left(\mathbf{x}_j - \mathbf{M}^{-1}\mathbf{W}^{\mathrm{T}}\mathbf{t}_j\right)^{\mathrm{T}}(\sigma^{-2}\mathbf{M})\left(\mathbf{x}_j - \mathbf{M}^{-1}\mathbf{W}^{\mathrm{T}}\mathbf{t}_j\right)\right), \end{aligned} \tag{13}$$

where a matrix $\mathbf{M}$ is given by $\mathbf{M} = \mathbf{W}^{\mathrm{T}}\mathbf{W} + \sigma^2\mathbf{I}$.

2.2.1.3 Maximum Likelihood Estimates

For the derivation of $\mathbf{W}_{\mathrm{ML}}$ and σ^2_{ML}, i.e., the MLE of $\mathbf{W}$ and σ^2, the log-likelihood function of $\mathbf{t}_j$ is constructed as

$$\mathcal{L}(\mathbf{W}, \sigma^2) = \sum_{j=1}^{N} \ln p(\mathbf{t}_j; \mathbf{W}, \sigma^2) = -\frac{Nd}{2}\ln(2\pi) - \frac{N}{2}\ln|\mathbf{C}| - \frac{N}{2}\operatorname{tr}\left(\mathbf{C}^{-1}\mathbf{S}\right), \tag{14}$$

with a sample covariance matrix $\mathbf{S}$ defined by

$$\mathbf{S} = \frac{1}{N}\mathbf{T}\mathbf{T}^{\mathrm{T}}.$$

According to the method of maximum likelihood, $\mathbf{W}_{\mathrm{ML}}$ and σ^2_{ML} can be found from the first derivative of $\mathcal{L}$ in Eq. (14) with respect to each parameter. Nevertheless, the maximum likelihood method fails to admit analytic solutions for $\mathbf{W}_{\mathrm{ML}}$ and σ^2_{ML} since the following stationary equations in Eq. (15) have no closed form solutions.

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mathbf{W}} = 0 \qquad &\Longrightarrow \qquad (\mathbf{C}^{-1}\mathbf{S} - \mathbf{I})\mathbf{C}^{-1}\mathbf{W} = 0, \\ \frac{\partial \mathcal{L}}{\partial \sigma^2} = 0 \qquad &\Longrightarrow \qquad \mathbf{I} - \mathbf{C}^{-1}\mathbf{S} = 0, \end{aligned} \tag{15}$$

Although Eq. (15) cannot help derive $\mathbf{W}_{\mathrm{ML}}$ and σ^2_{ML}, Tipping and Bishop[82] utilized the SVD of $\mathbf{W}$ such that $\mathbf{W} = \mathbf{Q}_1\mathbf{\Omega}\mathbf{Q}_2^{\mathrm{T}}$ in order to reveal that the MLEs will be eventually as follows:

$$\sigma^2_{\mathrm{ML}} = \frac{1}{d-q}\sum_{j=q+1}^{d}\lambda_j, \quad \text{and} \quad \mathbf{W}_{\mathrm{ML}} = \mathbf{V}_q(\mathbf{\Lambda}_q - \sigma^2\mathbf{I}_q)^{1/2}\mathbf{Q}_2^{\mathrm{T}}, \tag{16}$$

where $\mathbf{V}_q$ contains q eigenvectors of $\mathbf{S}$, $\mathbf{\Lambda}_q$ lists q eigenvalues of $\mathbf{S}$ corresponding to $\mathbf{V}_q$ in diagonal, and $\mathbf{Q}_2$ is an orthonormal matrix representing an arbitrary rotation.

In Eq. (16), σ^2_{ML} is the average of a $d-q$ number of abandoned eigenvalues of $\mathbf{S}$ indicating a projection error by whittling a dimension down from d to q. Whereas $\mathbf{W}_{\mathrm{ML}}$ in Eq. (16) is linearly transformed $\mathbf{V}_q$ through two additional operations: scaling by $(\mathbf{\Lambda}_q - \sigma^2\mathbf{I}_q)^{1/2}$ and rotation by $\mathbf{Q}_2$. Because of the additional linear transformations implicit in $\mathbf{W}_{\mathrm{ML}}$, a post-process is inevitable to retrieve $\mathbf{V}_q$ from $\mathbf{W}_{\mathrm{ML}}$. Note that the column vectors of $\mathbf{W}_{\mathrm{ML}}$ do span the same q dimensional subspace as does those of $\mathbf{V}_q$; however, they are *not* orthogonal.

2.2.1.4 EM Algorithm for PPCA

Although Tipping and Bishop[82] showed that $\mathbf{W}_{\mathrm{ML}}$ and σ^2_{ML} in Eq. (16) can be indirectly achievable with the SVD of $\mathbf{S}$, the EM algorithm is indispensable for PPCA to find its parameter MLEs if observations involve missing data. Dempster, Laird, and Rubin[10] developed the EM algorithm adumbrating several feasible applications of it, and later, Rubin and Thayer[64] articulated it for a factor analysis model pertinent to PPCA. Afterwards, Tipping and Bishop[82] capitalized on the EM algorithm formulating the EM-PCA to derive PPCA parameter MLEs. For parameter estimation, the EM algorithm iteratively yields

parameter MLEs alternating two steps: an E-step and an M-step. The E-step estimates unknown variables given current parameter estimates, and the subsequent M-step corrects the parameter estimates given the estimated variables in the previous E-step so as to maximize the expectation of a log-likelihood function. Literally, the EM algorithm requires the presence of hidden or missing data, yet applications of the EM algorithm are not limited only to incomplete observations since the existence of missing data is a virtual device to exploit the EM algorithm. Theoretically, the EM algorithm can always reach a local maximum of a likelihood function,[10] and particularly for PPCA, the EM-PCA can locate the global maximum of a likelihood function.[82]

For the application of the EM algorithm to PPCA, $\mathbf{t}_j$ and $\mathbf{x}_j$ are combined to generate a data set with inherent missing data due to $\mathbf{x}_j$. The probability model of the combined data set $(\mathbf{t}_j, \mathbf{x}_j)$ can be found as the joint distribution of $\mathbf{t}_j$ and $\mathbf{x}_j$ such that

$$\begin{aligned} p(\mathbf{t}_j, \mathbf{x}_j) &= p(\mathbf{t}_j|\mathbf{x}_j)p(\mathbf{x}_j) \\ &= (2\pi\sigma^2)^{-d/2} \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{t}_j - \mathbf{W}\mathbf{x}_j\|^2\right) (2\pi)^{-q/2} \exp\left(-\frac{1}{2}\|\mathbf{x}_j\|^2\right). \end{aligned} \tag{17}$$

Expectation Step If observations contain no missing data, a latent variable $\mathbf{x}_j$ is the only unknown data to be estimated. From the posterior probability of $\mathbf{x}_j$ in Eq. (13) such that

$$p(\mathbf{x}_j|\mathbf{t}_j) = \mathcal{N}(\mathbf{M}^{-1}\mathbf{W}^{\mathrm{T}}\mathbf{t}_j, \sigma^2\mathbf{M}^{-1})$$

the expectation of $\mathbf{x}_j$ is determined as $\langle \mathbf{x}_j \rangle = \mathbf{M}^{-1}\mathbf{W}^{\mathrm{T}}\mathbf{t}_j$. However, if observations have some missing data, both $\mathbf{x}_j$ and $\mathbf{t}_j$ are the two unknown variables to be estimated. Therefore, along with the evaluation of $\langle \mathbf{x}_j \rangle$, the E-step necessitates the evaluation of $\langle \mathbf{t}_j \rangle$ such that $\langle \mathbf{t}_j \rangle = \mathbf{W}\mathbf{x}_j$ from the conditional probability of $\mathbf{t}_j$ in Eq. (13). In a matrix form, both $\langle \mathbf{x}_j \rangle$ and $\langle \mathbf{t}_j \rangle$ can be expressed as

$$\langle \mathbf{X} \rangle = \mathbf{M}^{-1}\mathbf{W}^{\mathrm{T}}\mathbf{T}, \quad \text{and} \quad \langle \mathbf{T} \rangle = \mathbf{W}\mathbf{X}.$$

with $\mathbf{X} = \{\mathbf{x}_j\}_{j=1}^N \in \mathbb{R}^{q \times N}$ and $\mathbf{T} = \{\mathbf{t}_j\}_{j=1}^N \in \mathbb{R}^{d \times N}$. Note that an E-step normally evaluates only $\langle \mathbf{x}_j \rangle$, and when it evaluates both $\langle \mathbf{x}_j \rangle$ and $\langle \mathbf{t}_j \rangle$, it is called a generalized E-step.[63]

Maximization Step Given the joint probability distribution of $(\mathbf{t}_j, \mathbf{x}_j)$ in Eq. (17), the log-likelihood of $(\mathbf{t}_j, \mathbf{x}_j)$ is evaluated as

$$\begin{aligned}\mathcal{L}_C(\mathbf{W}, \sigma^2) &= \sum_{j=1}^{N} \ln p(\mathbf{t}_j, \mathbf{x}_j; \mathbf{W}, \sigma^2) \\ &= -\frac{N}{2}(d+q)\ln(2\pi) - \frac{Nd}{2}\ln(\sigma^2) - \frac{1}{2\sigma^2}\sum_{j=1}^{N}\|\mathbf{t}_j - \mathbf{W}\mathbf{x}_j\|^2 - \frac{1}{2}\sum_{j=1}^{N}\|\mathbf{x}_j\|^2,\end{aligned} \tag{18}$$

and its expectation is found as follows:

$$\begin{aligned}\langle\mathcal{L}_C\rangle = &-\frac{N}{2}(d+q)\ln(2\pi) - \frac{Nd}{2}\ln(\sigma^2) \\ &- \frac{1}{2\sigma^2}\operatorname{tr}\left(\mathbf{T}^{\mathrm{T}}\mathbf{T} - 2\langle\mathbf{X}\rangle^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\mathbf{T} + \mathbf{W}^{\mathrm{T}}\mathbf{W}\langle\mathbf{X}\mathbf{X}^{\mathrm{T}}\rangle\right) - \frac{1}{2}\operatorname{tr}\left(\langle\mathbf{X}\mathbf{X}^{\mathrm{T}}\rangle\right),\end{aligned} \tag{19}$$

where

$$\langle\mathbf{X}\rangle = \mathbf{M}^{-1}\mathbf{W}^{\mathrm{T}}\mathbf{T}, \quad \text{and} \quad \langle\mathbf{X}\mathbf{X}^{\mathrm{T}}\rangle = \sigma^2\mathbf{M}^{-1} + \langle\mathbf{X}\rangle\langle\mathbf{X}\rangle^{\mathrm{T}}. \tag{20}$$

Finally, parameter estimates that maximize $\langle\mathcal{L}_C\rangle$ in Eq. (19) can be derived from the first derivatives of $\langle\mathcal{L}_C\rangle$ with respect to $\mathbf{W}$ and σ^2 as below.

$$\begin{aligned}\frac{\partial\langle\mathcal{L}_C\rangle}{\partial\mathbf{W}} = 0 \qquad &\Longrightarrow \qquad \widetilde{\mathbf{W}} = \mathbf{T}\langle\mathbf{X}\rangle^{\mathrm{T}}\langle\mathbf{X}\mathbf{X}^{\mathrm{T}}\rangle^{-1}, \\ \frac{\partial\langle\mathcal{L}_C\rangle}{\partial\sigma^2} = 0 \qquad &\Longrightarrow \qquad \widetilde{\sigma}^2 = \frac{1}{Nd}\operatorname{tr}\left(\mathbf{T}^{\mathrm{T}}\mathbf{T} - 2\langle\mathbf{X}\rangle^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\mathbf{T} + \mathbf{W}^{\mathrm{T}}\mathbf{W}\langle\mathbf{X}\mathbf{X}^{\mathrm{T}}\rangle\right),\end{aligned} \tag{21}$$

EM-PCA After all, the previously E- and M-steps constitute the EM-PCA such that

$$\text{Generalized E-step:} \qquad \langle\mathbf{X}\rangle = \left(\mathbf{W}^{\mathrm{T}}\mathbf{W}\right)^{-1}\mathbf{W}^{\mathrm{T}}\mathbf{T}, \tag{22a}$$

$$\langle\mathbf{T}\rangle = \mathbf{W}\mathbf{X}, \tag{22b}$$

$$\text{M-step:} \qquad \widetilde{\mathbf{W}} = \mathbf{T}\mathbf{X}^{\mathrm{T}}\left(\mathbf{X}\mathbf{X}^{\mathrm{T}}\right)^{-1}, \tag{22c}$$

under a zero-noise limit, i.e., $\lim\sigma^2 \to 0$. In addition to the EM-PCA in Eq. (22), which presumes mean-centered data, the EM-PCA can easily expand to incorporate a mean estimation with a sample mean $\overline{\mathbf{t}}$ in the M-step as follows.

$$\text{Generalized E-step:} \qquad \langle\mathbf{X}\rangle = \left(\mathbf{W}^{\mathrm{T}}\mathbf{W}\right)^{-1}\mathbf{W}^{\mathrm{T}}\left(\mathbf{T} - \overline{\mathbf{T}}\right), \tag{23a}$$

$$\langle\mathbf{T}\rangle = \mathbf{W}\mathbf{X} + \overline{\mathbf{T}}, \tag{23b}$$

$$\text{M-step:} \qquad \widetilde{\mathbf{W}} = \left(\mathbf{T} - \overline{\mathbf{T}}\right)\mathbf{X}^{\mathrm{T}}\left(\mathbf{X}\mathbf{X}^{\mathrm{T}}\right)^{-1}, \tag{23c}$$

$$\widetilde{\boldsymbol{\mu}} = \overline{\mathbf{t}}, \tag{23d}$$

where $\overline{\mathbf{T}} = \bar{\mathbf{t}}\mathbf{1}_N^{\mathrm{T}} \in \mathbb{R}^{d\times N}$, which contains $\bar{\mathbf{t}}$ in column. Note that if observations include no missing data, both the EM-PCA algorithms in Eq. (22) and Eq. (23) can ignore the evaluation of $\langle \mathbf{T} \rangle$ in their E-step. Since $\mathbf{W}$ is the parameter related to $\mathbf{V}_q$ as shown in Eq. (16), the EM-PCA can generate $\mathbf{V}_q$ after distilling $\mathbf{W}_{\mathrm{ML}}$.

2.2.1.5 Limitations of the EM-PCA

As gappy POD is susceptible to the two special missing data types, illustrated in Figure 6, so is the EM-PCA. For the first type in Figure 6(a), which is absent of an entire column of a data set, unlike gappy POD, the EM-PCA does not break down though an absent column will be trivially filled with a sample mean. Similarly, for the other type in Figure 6(b), which misses an entire row of a data set, the EM-PCA cannot recover a missing row like gappy POD because a sample mean is unavailable for the completely absent row.

CHAPTER III

COMPARATIVE STUDY I: EM-PCA VS. POD

3.1 Theoretical Equivalence

3.1.1 Standard POD and PPCA

In Section 2.2.1.3, the MLE of $\mathbf{W}$ can be found with the first derivative of the log-likelihood function $\mathcal{L}$ in Eq. (14) with respect to $\mathbf{W}$ such that

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}} = N\left[\mathrm{tr}\left(\mathbf{C}^{-1}\mathbf{S}\mathbf{C}^{-1}\mathbf{W} - \mathbf{C}^{-1}\mathbf{W}\right)\right] = \left(\mathbf{S}\mathbf{C}^{-1} - \mathbf{I}_d\right)\mathbf{W} = 0,$$

which yields

$$\mathbf{S}\mathbf{C}^{-1}\mathbf{W} = \mathbf{W}. \tag{24}$$

Tipping and Bishop[82] showed that $\mathbf{W}$ satisfies Eq. (24) at the following three conditions: (i) $\mathbf{W} = 0$, (ii) $\mathbf{S} = \mathbf{C}$, and (iii) $\mathbf{S}\mathbf{C}^{-1}\mathbf{W} = \mathbf{W}$. Among the three solutions of Eq. (24), the first solution is trivial, and the second solution indicates that a sample covariance matrix $\mathbf{S}$ is exactly the same as the model covariance matrix $\mathbf{C}$, which is only admissible if $q \geq \mathrm{rank}(\mathbf{S})$. Unlike the first two solutions, the last solution is conducive to revealing that the factor-loading matrix $\mathbf{W}$ of PPCA is related to the POD basis $\mathbf{V}$ of the standard POD.

Let the SVD of $\mathbf{W}$ be $\mathbf{Q}_1\mathbf{\Omega}\mathbf{Q}_2^{\mathrm{T}}$ where $\mathbf{Q}_1$ is a d-by-q orthogonal matrix, $\mathbf{\Omega}$ is a q-by-q diagonal matrix, and $\mathbf{Q}_2$ is a q-by-q orthogonal matrix. First, $\mathbf{C}^{-1}\mathbf{W}$ in the left-hand side (LHS) of Eq. (24) can be expanded with the SVD of $\mathbf{W}$ as follows:

$$\begin{aligned}\mathbf{C}^{-1}\mathbf{W} &= \left(\mathbf{W}\mathbf{W}^{\mathrm{T}} + \sigma^2\mathbf{I}_d\right)^{-1}\mathbf{W} = \left(\mathbf{Q}_1\mathbf{\Omega}^2\mathbf{Q}_1^{\mathrm{T}} + \sigma^2\mathbf{I}_d\right)^{-1}\mathbf{Q}_1\mathbf{\Omega}\mathbf{Q}_2^{\mathrm{T}} \\ &= \left(\mathbf{Q}_1(\mathbf{\Omega}^2 + \sigma^2\mathbf{I}_q)\mathbf{Q}_1^{\mathrm{T}}\right)^{-1}\mathbf{Q}_1\mathbf{\Omega}\mathbf{Q}_2^{\mathrm{T}} = \mathbf{Q}_1\left(\mathbf{\Omega}^2 + \sigma^2\mathbf{I}_q\right)^{-1}\mathbf{\Omega}\mathbf{Q}_2^{\mathrm{T}}.\end{aligned}$$

With the above rephrased $\mathbf{C}^{-1}\mathbf{W}$, Eq. (24) reduces to

$$\mathbf{S}\mathbf{Q}_1\left(\mathbf{\Omega}^2 + \sigma^2\mathbf{I}_q\right)^{-1}\mathbf{\Omega}\mathbf{Q}_2^{\mathrm{T}} = \mathbf{Q}_1\mathbf{\Omega}\mathbf{Q}_2^{\mathrm{T}},$$

which can be further transformed into the following familiar form of either EVD or SVD of $\mathbf{S}$ such that

$$\mathbf{S} = \mathbf{Q}_1\left(\mathbf{\Omega}^2 + \sigma^2\mathbf{I}_q\right)\mathbf{Q}_1^{\mathrm{T}} \quad \text{or} \quad \mathbf{S}\mathbf{Q}_1 = \mathbf{Q}_1\left(\mathbf{\Omega}^2 + \sigma^2\mathbf{I}_q\right).$$

As the previous Section 2.1.1.1 described that the standard POD formulation addresses the following EVD or SVD of $\mathbf{S}$ such that

$$\mathbf{SV} = \mathbf{V}\boldsymbol{\Lambda} \quad \text{or} \quad \mathbf{S} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{\mathrm{T}},$$

it turns out that $\mathbf{Q}_1$ of PPCA corresponds to $\mathbf{V}$ of the standard POD, as does $\left(\boldsymbol{\Omega}^2 + \sigma^2\mathbf{I}_q\right)$ to $\boldsymbol{\Lambda}$. Therefore, $\mathbf{Q}_1$ is identical to $\mathbf{V}_q$, the first leading q eigenvectors in $\mathbf{V}$, and likewise the diagonal matrix $\left(\boldsymbol{\Omega}^2 + \sigma^2\mathbf{I}_q\right)$ is the same as $\boldsymbol{\Lambda}_q$, the first q eigenvalues in $\boldsymbol{\Lambda}$, which determines the diagonal elements of $\boldsymbol{\Omega}$ as follows:

$$\omega_j = \sqrt{\lambda_j - \sigma^2} \quad \text{where} \quad j = 1, \ldots, q.$$

However, due to arbitrary rotation $\mathbf{Q}_2$ that is indeterminate, as noted in the earlier Section 2.2.1.3, the analytic solution of $\mathbf{W}$ cannot be found by the maximum likelihood method.

Similarly, the MLE of σ^2 can be derived from the first derivative of the log-likelihood function $\mathcal{L}$ in Eq. (14) with respect to σ^2 such that

$$\frac{\partial \mathcal{L}}{\partial \sigma^2} = -\frac{N}{2}\left[\mathrm{tr}\left(\mathbf{C}^{-1}\left(\mathbf{I}_d - \mathbf{C}^{-1}\mathbf{S}\right)\right)\right] = 0,$$

which yields a stationary condition for σ^2 as follows:

$$\mathbf{C} = \mathbf{S} \quad \Longrightarrow \quad \sigma^2\mathbf{I}_d = \mathbf{S} - \mathbf{W}\mathbf{W}^{\mathrm{T}}. \tag{25}$$

For the right-hand side (RHS) of Eq. (25), the SVD of $\mathbf{S}$ can be decomposed such that

$$\mathbf{S} = \left[\begin{array}{c:c} \mathbf{V}_q & \mathbf{V}_{d-q} \end{array}\right] \left[\begin{array}{c:c} \boldsymbol{\Lambda}_q & \mathbf{0} \\ \hdashline \mathbf{0} & \boldsymbol{\Lambda}_{d-q} \end{array}\right] \left[\begin{array}{c} \mathbf{V}_q^{\mathrm{T}} \\ \hdashline \mathbf{V}_{d-q}^{\mathrm{T}} \end{array}\right],$$

and $\mathbf{W}\mathbf{W}^{\mathrm{T}}$ is rephrased with its SVD and then decomposed as

$$\mathbf{W}\mathbf{W}^{\mathrm{T}} = \mathbf{Q}_1\boldsymbol{\Omega}^2\mathbf{Q}_1^{\mathrm{T}} = \mathbf{V}_q\left(\boldsymbol{\Lambda}_q - \sigma^2\mathbf{I}_q\right)\mathbf{V}_q^{\mathrm{T}} = \left[\begin{array}{c:c} \mathbf{V}_q & \mathbf{0}_{d-q} \end{array}\right] \left[\begin{array}{c:c} \boldsymbol{\Lambda}_q - \sigma^2\mathbf{I}_q & \mathbf{0} \\ \hdashline \mathbf{0} & \mathbf{0}_{d-q} \end{array}\right] \left[\begin{array}{c} \mathbf{V}_q^{\mathrm{T}} \\ \hdashline \mathbf{0}_{d-q}^{\mathrm{T}} \end{array}\right]$$

with the previous findings of $\mathbf{Q}_1 = \mathbf{V}_q$ and $\boldsymbol{\Omega}^2 + \sigma^2\mathbf{I}_q = \boldsymbol{\Lambda}_q$. Since the LHS of Eq. (25) can be expressed as $\sigma^2\mathbf{I}_d = \mathbf{V}\left(\sigma^2\mathbf{I}_d\right)\mathbf{V}^{\mathrm{T}}$, it can be also decomposed as the below.

$$\left[\begin{array}{c:c} \mathbf{V}_q & \mathbf{V}_{d-q} \end{array}\right] \left[\begin{array}{c:c} \sigma^2\mathbf{I}_q & \mathbf{0} \\ \hdashline \mathbf{0} & \sigma^2\mathbf{I}_{d-q} \end{array}\right] \left[\begin{array}{c} \mathbf{V}_q^{\mathrm{T}} \\ \hdashline \mathbf{V}_{d-q}^{\mathrm{T}} \end{array}\right] = \left[\begin{array}{c:c} \mathbf{V}_q & \mathbf{V}_{d-q} \end{array}\right] \left[\begin{array}{c:c} \sigma^2\mathbf{I}_q & \mathbf{0} \\ \hdashline \mathbf{0} & \boldsymbol{\Lambda}_{d-q} \end{array}\right] \left[\begin{array}{c} \mathbf{V}_q^{\mathrm{T}} \\ \hdashline \mathbf{V}_{d-q}^{\mathrm{T}} \end{array}\right].$$

Finally, with the decomposed RHS and LHS of Eq. (25), Eq. (25) results in the following:

$$\mathbf{V}_{d-q}\left(\sigma^2\mathbf{I}_{d-q}\right)\mathbf{V}_{d-q}^{\mathrm{T}} = \mathbf{V}_{d-q}\left(\mathbf{\Lambda}_{d-q}\right)\mathbf{V}_{d-q}^{\mathrm{T}} \quad \Longrightarrow \quad \sigma^2\mathbf{I}_{d-q} = \mathbf{\Lambda}_{d-q},$$

revealing the maximum likelihood σ^2 as

$$\sigma^2 = \frac{1}{d-q}\sum_{j=q+1}^{d}\lambda_j.$$

Note that when q reaches to d, PPCA becomes equivalent to the standard POD, yielding $\mathbf{Q}_1 = \mathbf{V}$ and $\omega_j = \sqrt{\lambda_j}$.

3.2 Validation with Numerical Simulations

In order to validate a POD basis indirectly obtained by the EM-PCA with that directly achieved by POD, Lee, Rallabhandi, and Mavris[35] applied the EM-PCA to two intact aerodynamic data sets. The simulation results were collected from two airfoil flowfield analyses with different flow solvers: one with a FPE solver, and the other with an Euler CFD solver. For the convergence monitoring of the EM-PCA, the following convergence criterion such that

$$|\mathrm{RMSR}^{(k)} - \mathrm{RMSR}^{(k-1)}| < 10^{-6}, \quad \text{or} \quad \frac{\mathrm{RMSR}^{(k)}}{\mathrm{RMSR}^{(1)}} < 10^{-6}$$

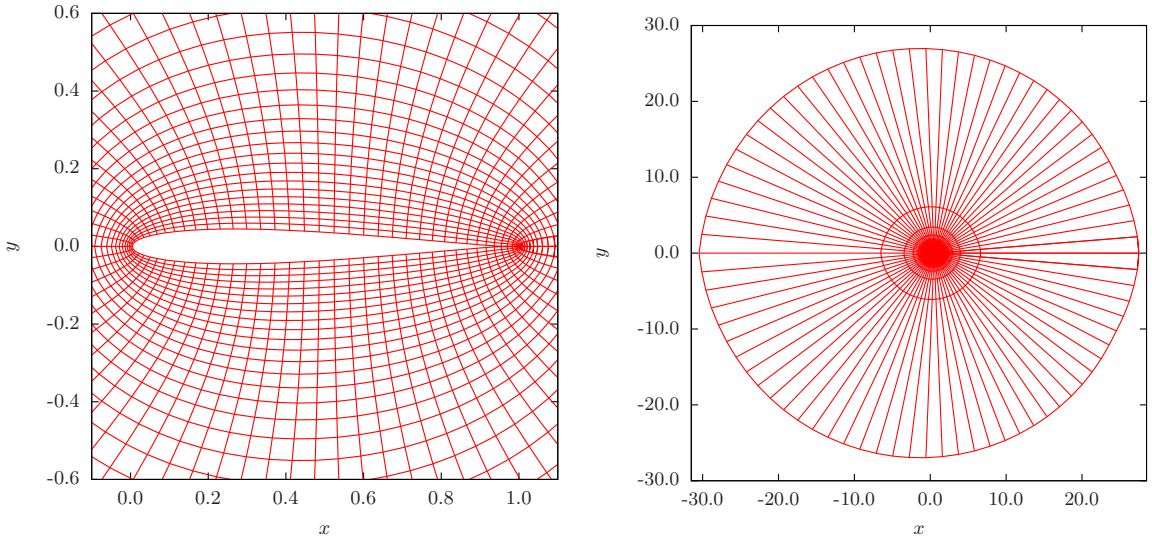
is utilized, and the root mean square residual (RMSR) of $\mathbf{W}$ is defined by

$$\mathrm{RMSR}^{(k)} = \sqrt{\frac{1}{dq}\sum_{j=1}^{N}\left\|\widetilde{\mathbf{w}}_j^{(k)} - \widetilde{\mathbf{w}}_j^{(k-1)}\right\|_{L^2}^2}.$$

3.2.1 Full Potential Equations

For an FPE solver, this research employed a transonic two-dimensional FPE solver implemented by Malone and Sankar[46] based on the rotated difference scheme by Jameson.[23] The FPE solver calculates surface pressure coefficients over the NACA 0012 airfoil whose discretized analysis domain is depicted in Figure 7. In order to generate sample data, Lee, Rallabhandi, and Mavris[35] produced a total of 50 snapshots by varying two parameters: a Mach number ($0.3 \sim 0.6$) and an angle of attack ($0^\circ \sim 2^\circ$). With the help of JMP software,[24] analysis snapshots are populated according to a Latin hypercube design (LHD), a

space filling design[65,75] tailored for computer experiments in design of experiments (DoE). Through FPE simulations, a 79-by-50 data set is produced, and the the numbe of modes q for the EM-PCA is set to five based on the eigenspectrum analysis of the data set; five modes are enough to delineate the total variations observed in the FPE simulation data set. Due to a relatively small data set size, the EM-PCA with $q = 5$ converges in a few iterations as shown in Figure 8.



(a) Zoom-in (b) Zoom-out

Figure 7: Computational domain of FPE analysis

Although the number of surface grid points is 79, the true dimension of the FPE data set is much lower than 79; according to the normalized eigenspectrum in Figure 9, four modes delineate 99.99% of variations observed in the data set. For instance, the first mode captures 93.6% of variations in the data set, and along with the second mode, the first two dominant modes together describe 99.7% of the variations. Overall, Figure 9 demonstrates that the EM-PCA can find a normalized eigenspectrum identical to those obtained by the snapshot POD method. Similar to the eigenspectrum validation in Figure 9, the mode validation in Figure 10 illustrates that the modes obtained by the EM-PCA perfectly align with those achieved by the snapshot POD. Note that the quality of modes obtained by the EM-PCA does not degrade even for insignificant modes, e.g., the third and the fourth modes shown in Figures 10(c) and 10(d), respectively.

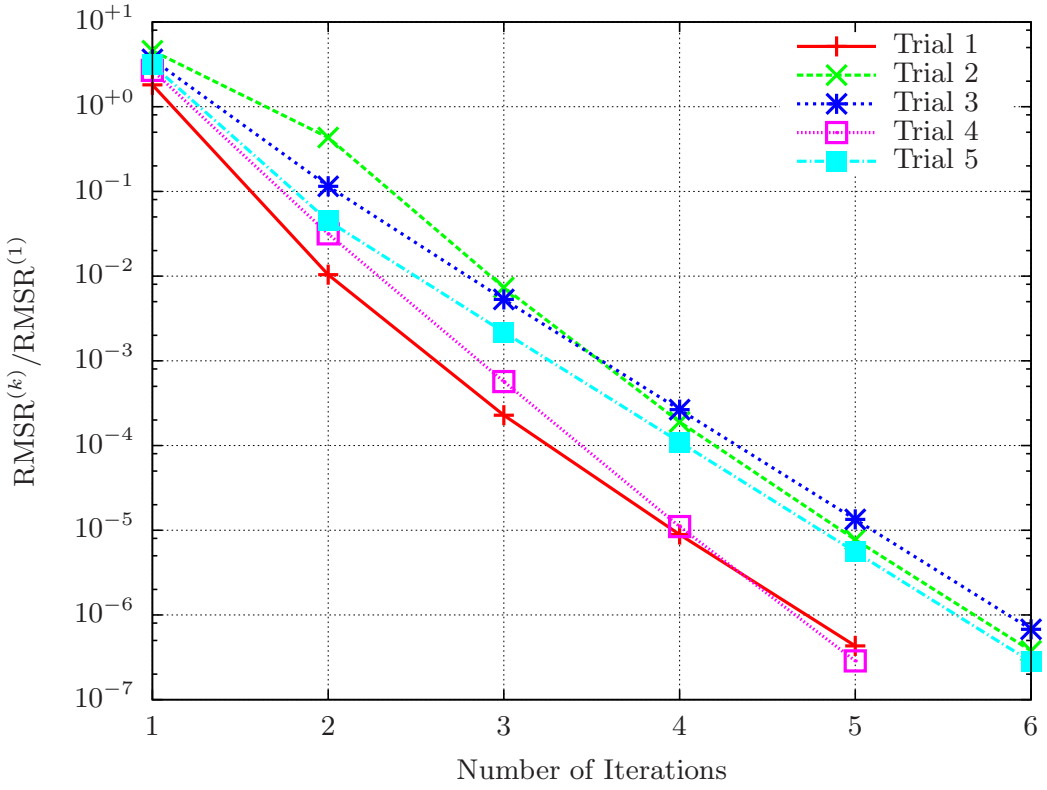


Figure 8: Convergence histories of the EM-PCA for FPE simulation data

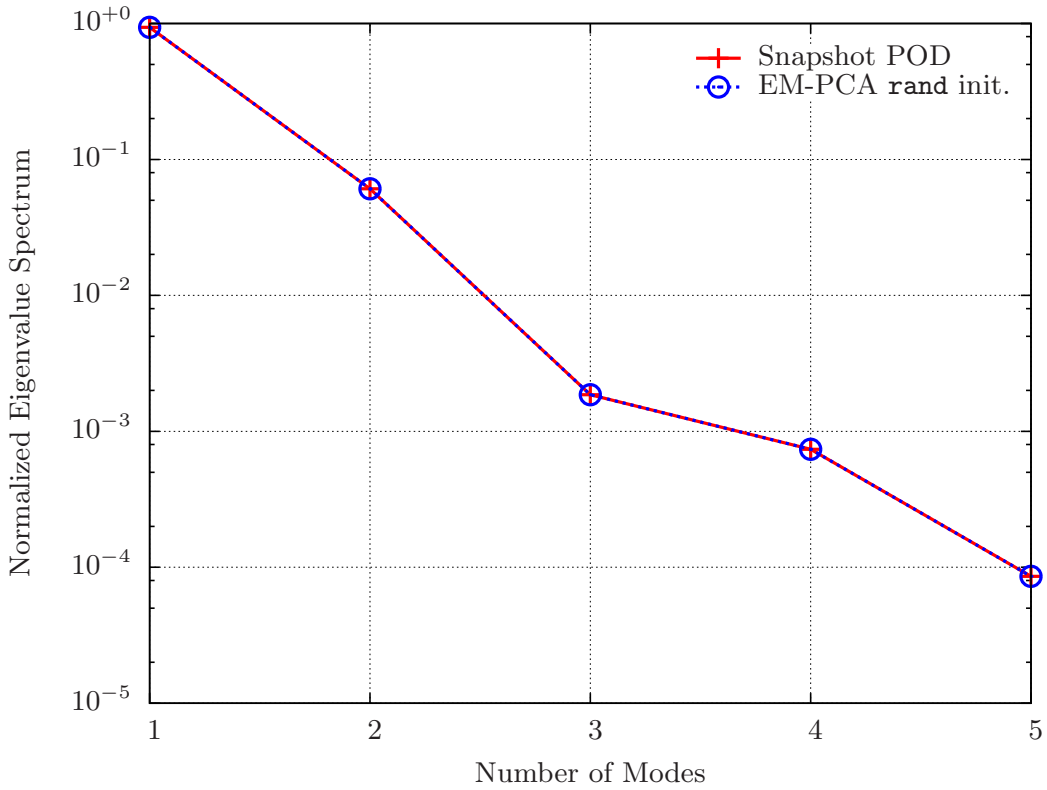
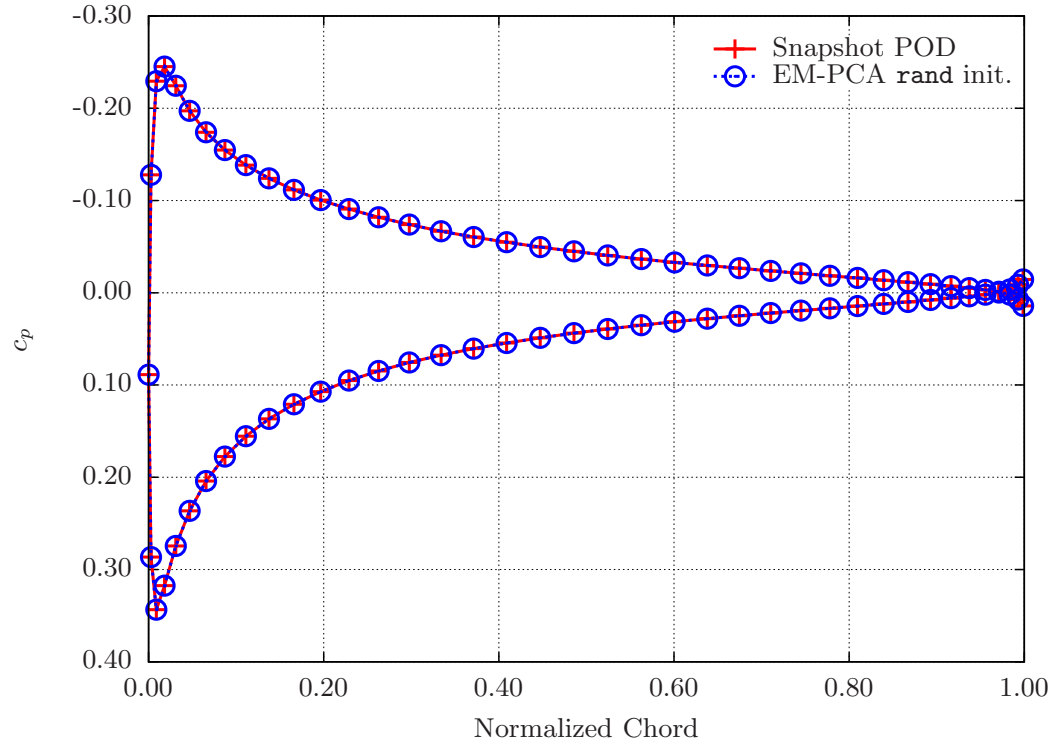
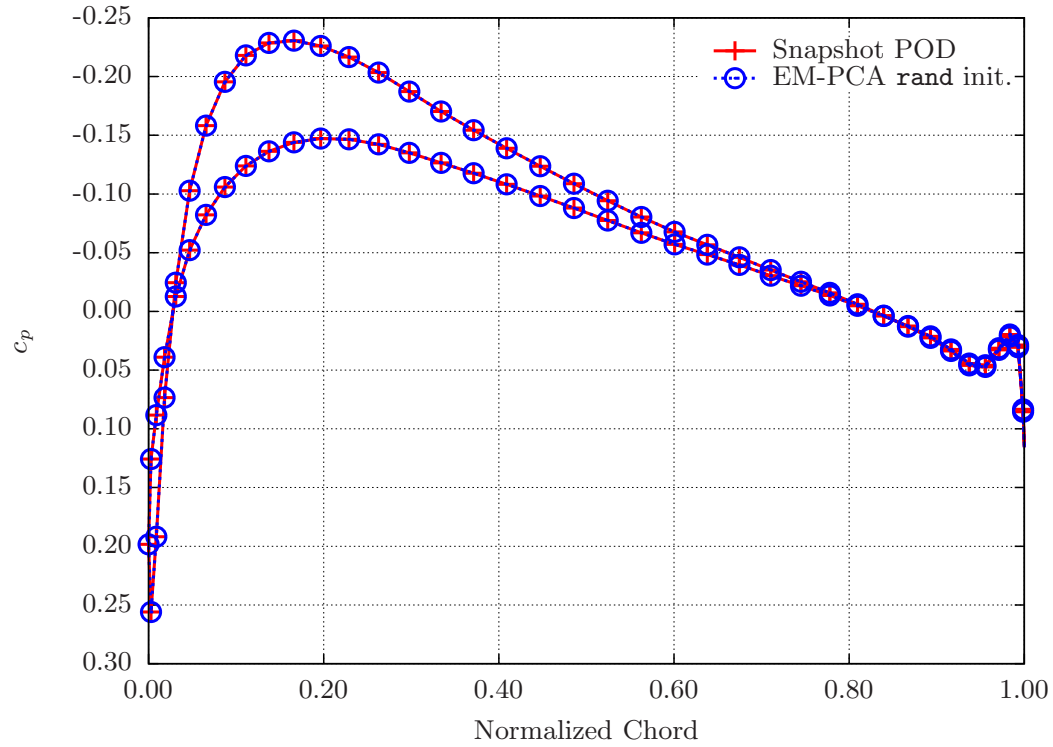


Figure 9: Eigenspectrum of FPE analysis

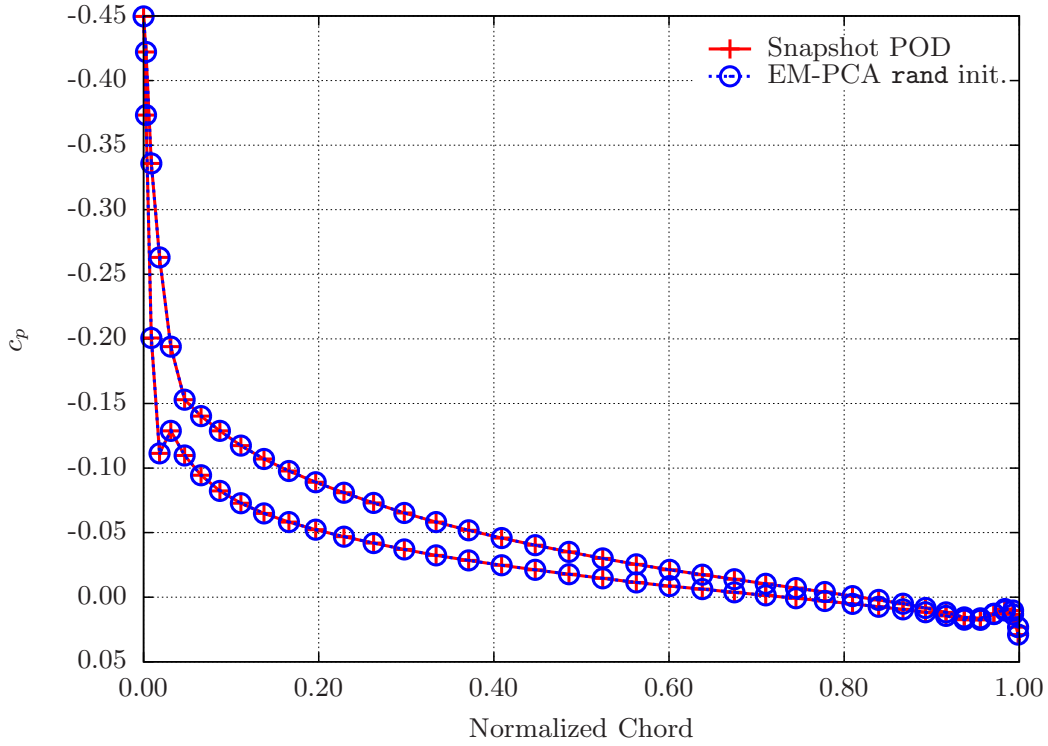


(a) 1st mode (93.65%)

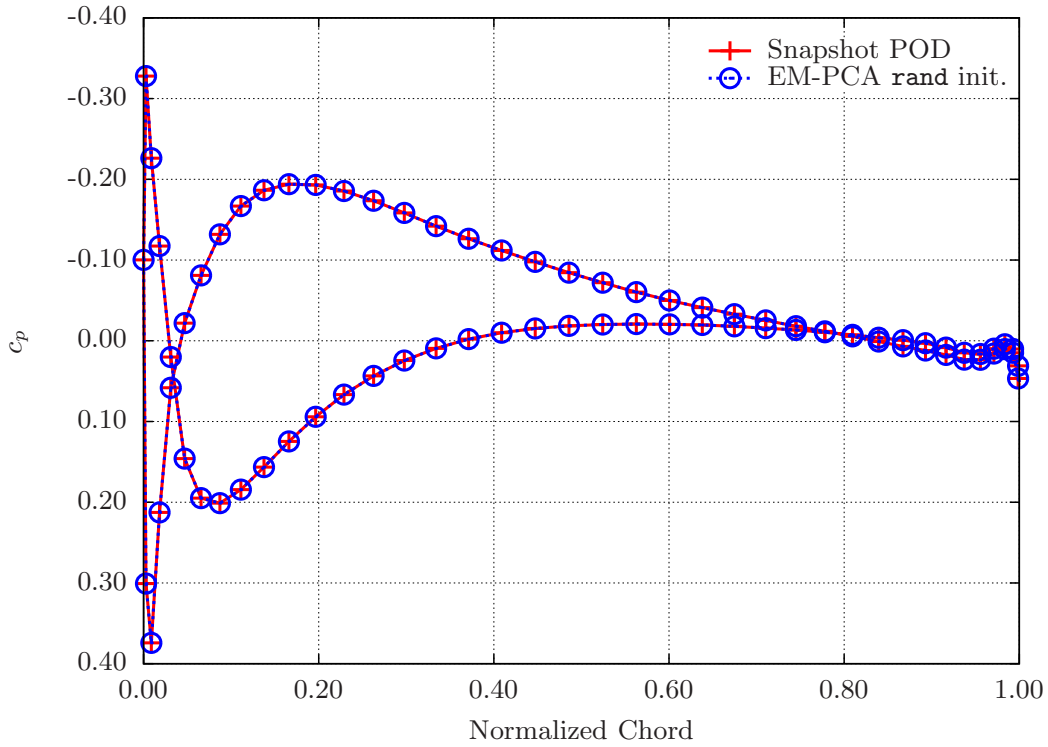


(b) 2nd mode (6.08%)

Figure 10: FPE surface pressure coefficient mode



(c) 3rd mode (0.19%)

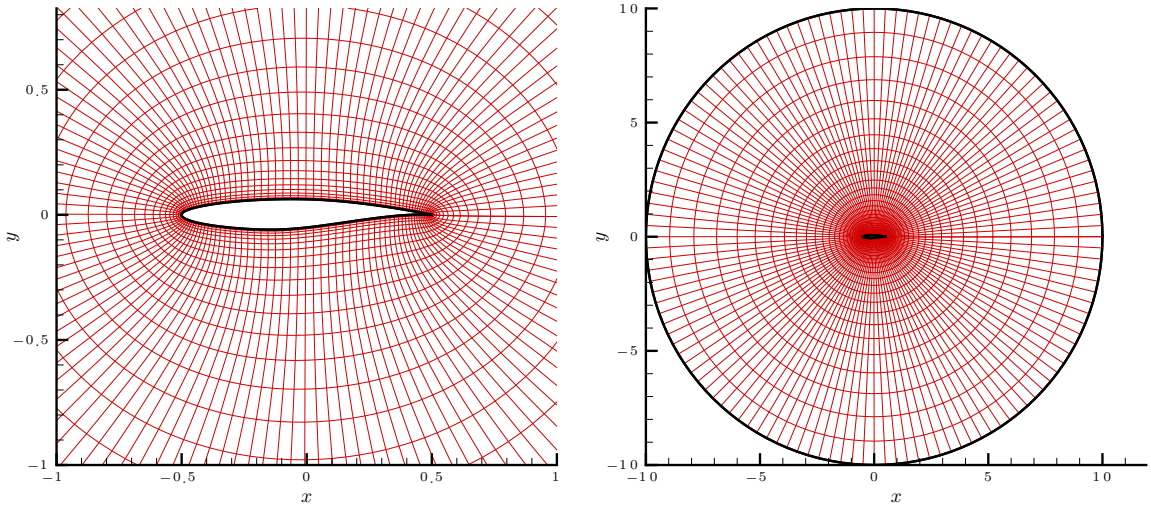


(d) 4th mode (0.07%)

Figure 10: FPE surface pressure coefficient mode

3.2.2 Euler Equations

For the second validation study, Lee, Rallabhandi, and Mavris[35] employed an Euler CFD solver developed by Min et al.[54] to analyze the flowfield around the RAE 2822 airfoil. As shown in Figure 11, the flowfield of interest is discretized into 129×33 grid points, and two parameters changed are a Mach number ($0.4 \sim 0.6$) and an angle of attack ($0^\circ \sim 2^\circ$). Similar to the previous validation study in Section 3.2.1, they utilized JMP[24] to efficiently generate 200 snapshots with a maximum entropy design, resulting in a 4257-by-200 pressure data set. In Figure 12, the validation of a normalized eigenspectrum shows that the EM-PCA with $q = 20$, whose convergence behavior is depicted in Figure 13, can find exactly the same eigenvalues as those obtained by the snapshot POD. Unlike the FPE analysis in Section 3.2.1, the normalized eigenspectrum in Figure 12 conveys that huge dimensionality reduction from 4257 to 10 is achieved since 10 modes are enough to capture overall variations observed in the Euler data set. Like the FPE analysis in Section 3.2.1, the first two dominant modes account for 99.3% of variations in the Euler data set, and the four leading modes delineate 99.77% of the variations. Analogous to the eigenspectrum validation in Section 3.2.1, the EM-PCA can produce pressure data modes identical to those by the snapshot POD, as illustrated with contour plots in Figure 14.



(a) Zoom-in (b) Zoom-out

Figure 11: Computational domain of Euler CFD analysis

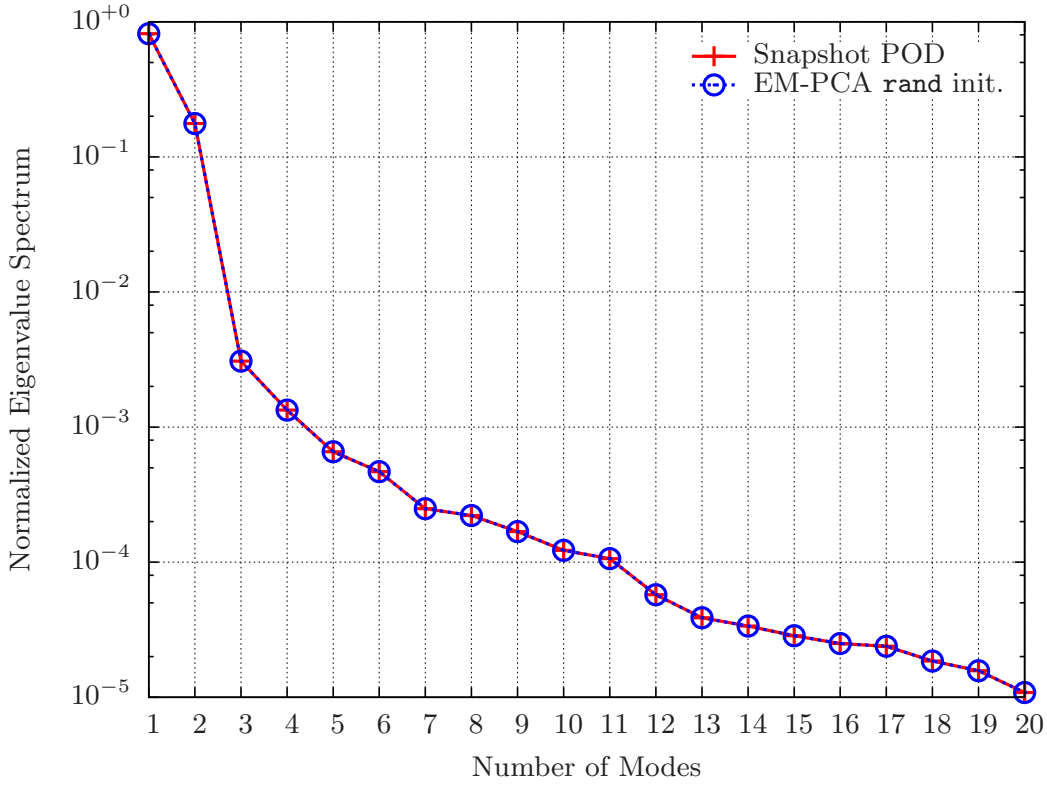


Figure 12: Eigenspectrum of the Euler airfoil pressure data

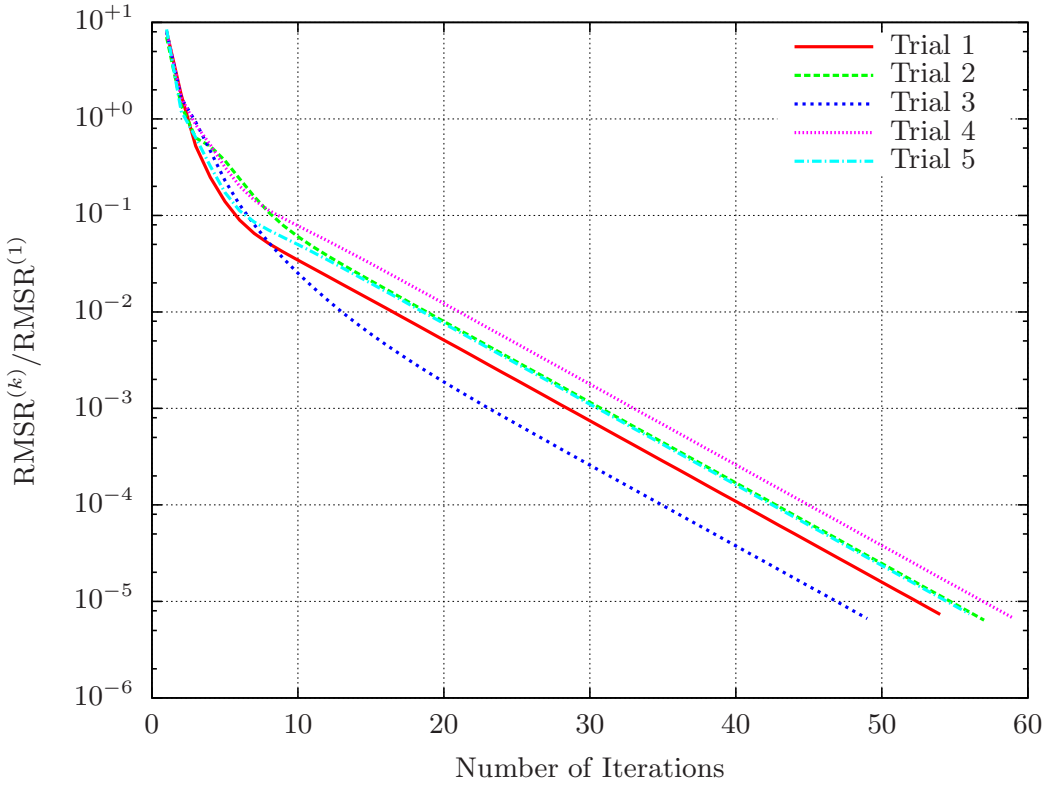
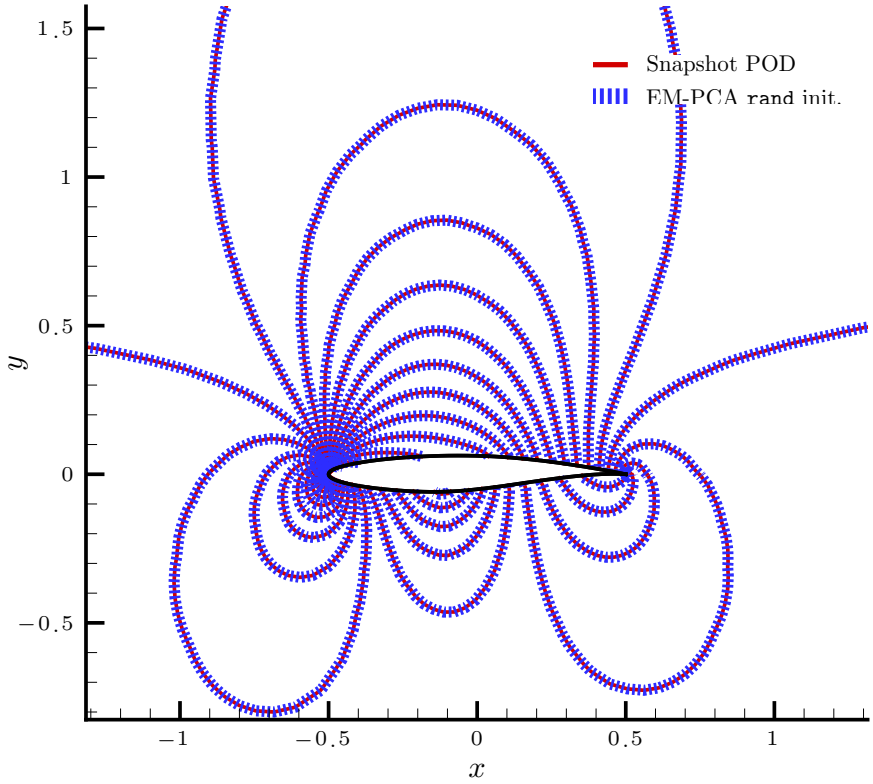
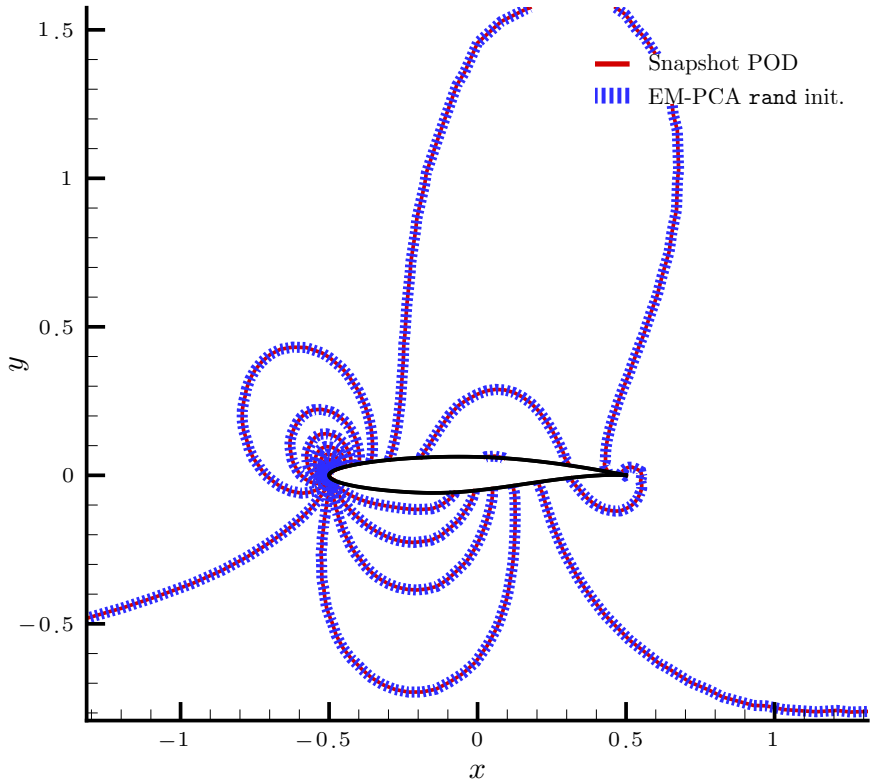


Figure 13: Convergence history of the EM-PCA for the Euler airfoil pressure data

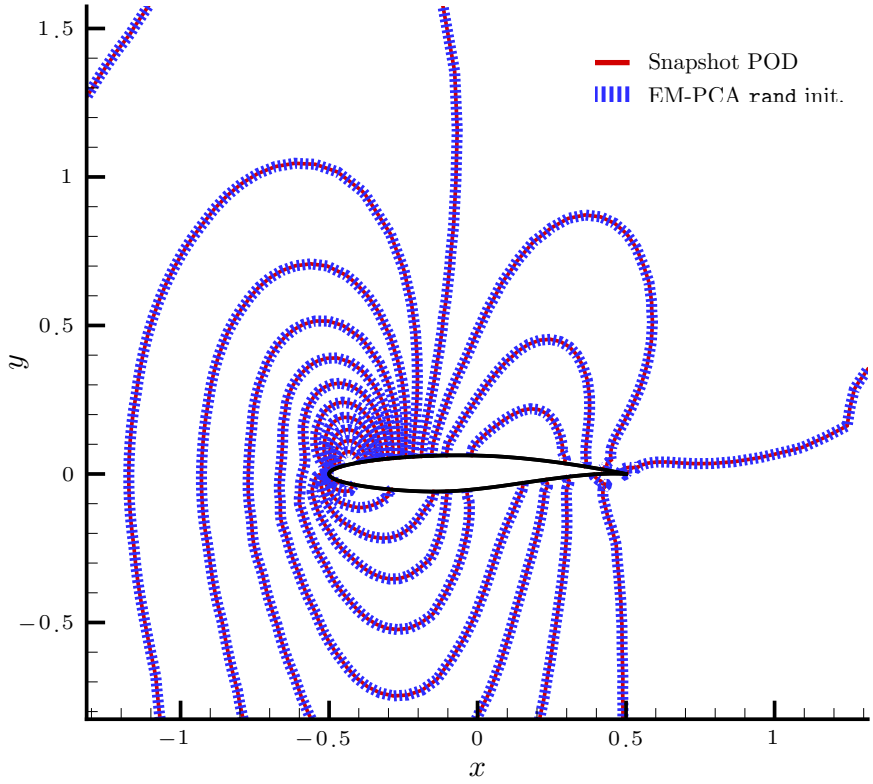


(a) 1st mode (81.68%)

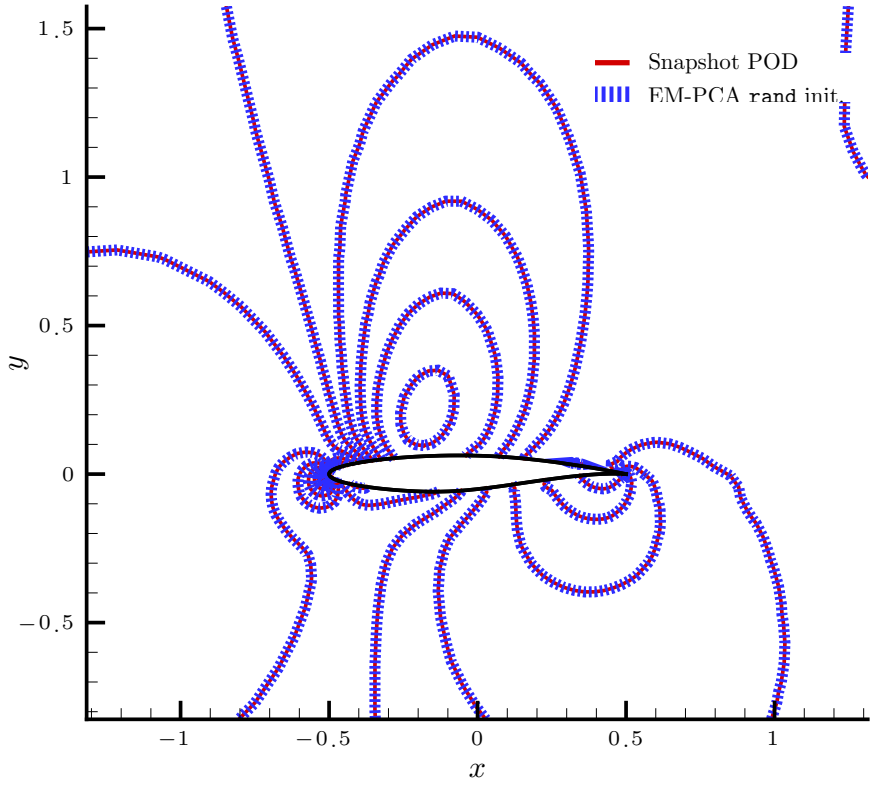


(b) 2nd mode (17.65%)

Figure 14: Contours of modes for the Euler airfoil pressure data



(c) 3rd mode (0.31%)



(d) 4th mode (0.13%)

Figure 14: Contours of modes for the Euler airfoil pressure data

3.3 Computational Efficiency Investigation

3.3.1 POD Methods and the EM-PCA

Table 1: Computational complexity comparison

	Standard POD	Snapshot POD	Snapshot POD w/ Lanczos	EM-PCA
Speed	$\mathcal{O}(Nd) + \mathcal{O}(d^3)$	$\mathcal{O}(Nd) + \mathcal{O}(N^3)$	$\mathcal{O}(qT^{\mathrm{op}} + q2N)$[a]	$\mathcal{O}(Ndq)$

[a] T^{op} is the complexity of multiplying T by a vector.[45]

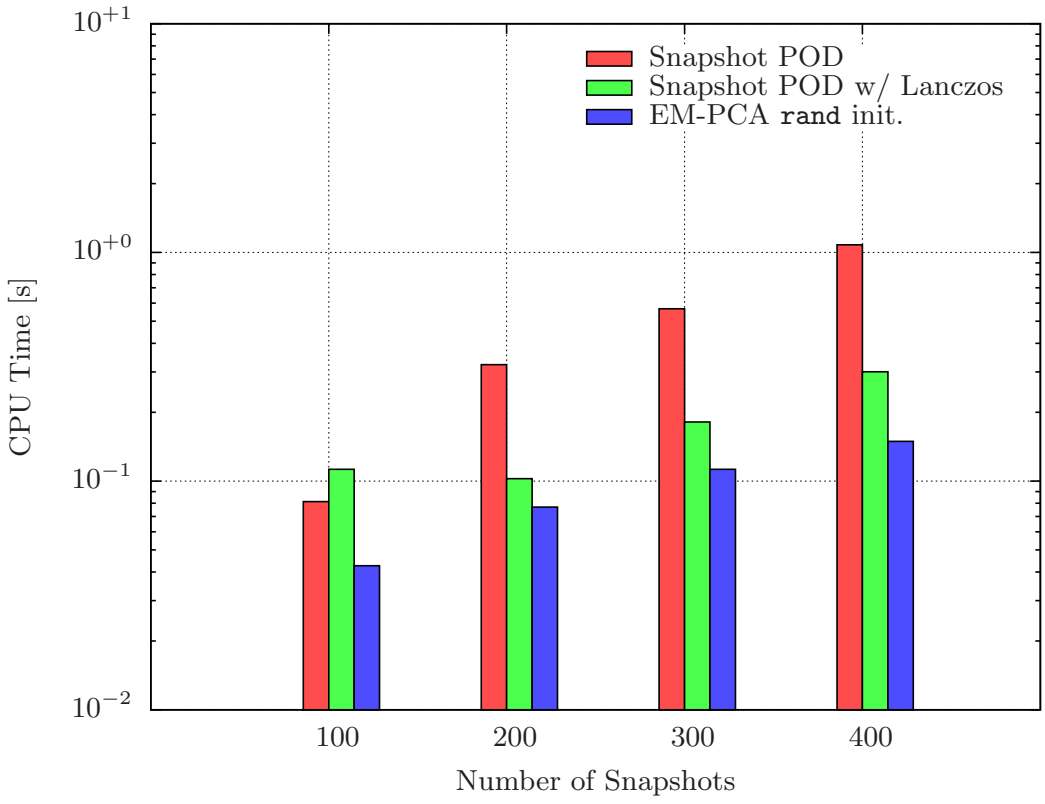
Based on the work of several researchers,[45,63,83] Table 1 summarizes the computational complexity of the two POD methods and the EM-PCA in terms of the big $\mathcal{O}$ notation. In addition, Table 1 also includes the computational complexity of the snapshot POD using the Lanczos algorithm since the algorithm is known as the most efficient scheme for extracting a low number of eigenvectors from a symmetric matrix. Note that both the EM-PCA and the Lanczos algorithm are iterative methods whose efficiency are affected by their convergence characteristics.

To begin with the POD methods, the POD algorithm is consist of two operations: the evaluation of a sample covariance matrix and the diagonalization of a sample covariance matrix with either EVD or SVD. According to Roweis,[63] the computational complexity of the standard POD is $\mathcal{O}(Nd^2)$ for its first step and $\mathcal{O}(d^3)$ for its second step, and likewise, that of the snapshot POD at its each step is $\mathcal{O}(dN^2)$ and $\mathcal{O}(N^3)$, respectively. Since high-fidelity aerodynamic analyses produce simulation data whose number of grid points d is enormous to whose number of snapshots N, the snapshot POD is a typical choice for efficient basis evaluations. However, as more design parameters are required for POD-based ROM, a snapshot ensemble size N has to grow because more snapshots are necessary for accurate modal coefficient approximation. This tendency of increasing N due to more parameters in POD-base ROM is expected to deteriorate the computational performance of the snapshot POD. In contrast, the computational complexity of the EM-PCA is $\mathcal{O}(dNq)$, which is linear to both d and N compared to the complexity of the standard and snapshot POD methods, both of whose complexities are cubic to d and N, respectively. Therefore,

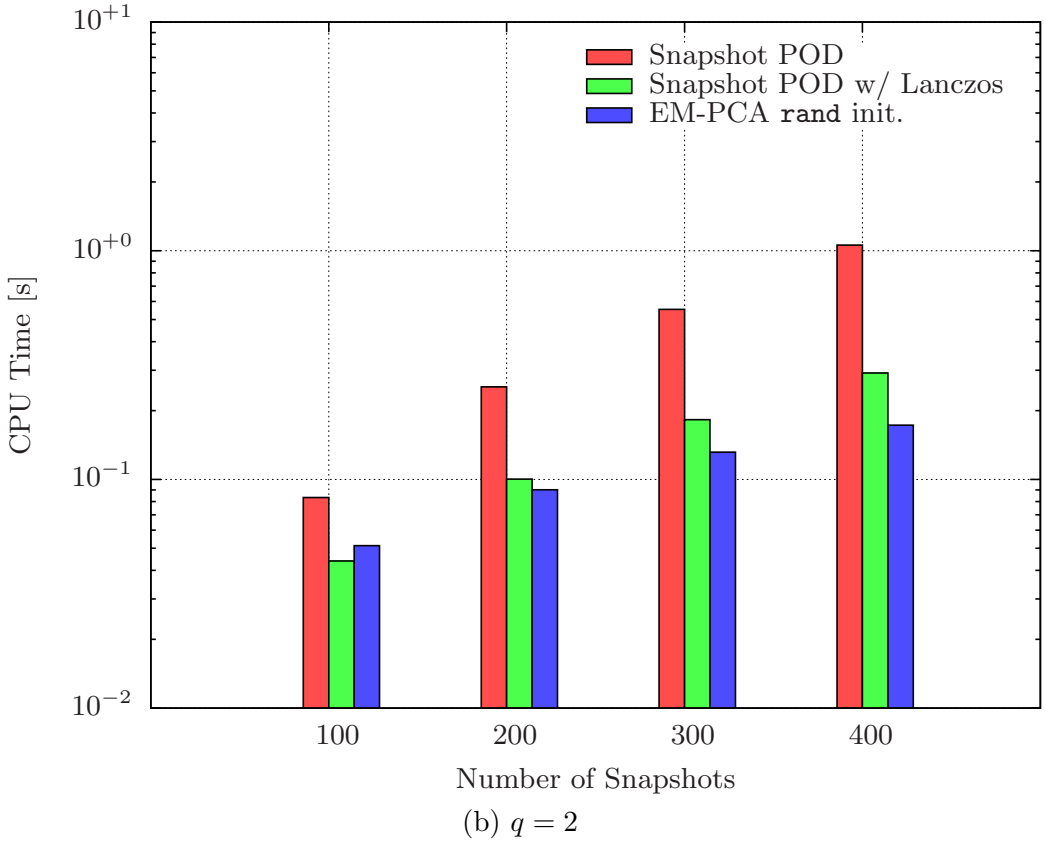
the EM-PCA is more scalable with the increase of either d or N than the standard and snapshot POD methods. Moreover, the complexity of the EM-PCA is linear to q, which is computationally beneficial when q is small. As an illustration, Roweis[63] demonstrated that the EM-PCA outperforms POD in the case of extracting only the first mode.

In order to examine the computational performance of the EM-PCA to the POD methods, this research measured their computational time, changing the number of modes q from one to four and the snapshot size N from 100 to 400 at intervals of 100. For sample data generation, the same two flow parameters as those for the earlier validation study in Section 3.2.2 were varied within the same ranges. Likewise, the previously utilized Euler CFD solver was employed again for the analysis of a flowfiled around the RAE 2822 airfoil at the various conditions of a Mach number and an angle of attack. Because of the random basis initialization of $\mathbf{W}$, the EM-PCA was run for 100 times, and then its computational time was averaged to mitigate random effect in measuring the performance of the EM-PCA. For numerical experiments, all tested algorithms were implemented in MATLAB, and the Lanczos algorithm is realized with a MATLAB function `eigs`, which relies on the well-known Fortran Library ARPACK.[38] Note that only the snapshot POD was tested since $d \gg N$.

Overall, Figure 15 delineates the computational time measurements of all the tested algorithms as N increases at each q value. First, in case of $q = 1$ in Figure 15(a), although the EM-PCA is slightly faster than the snapshot POD with the Lanczos algorithm, it is the most efficient than any other methods across all N values. As q grows from one to two, Figure 15(b) shows that the performance of the EM-PCA becomes almost identical to that of the snapshot POD with the Lanczos algorithm, but still the EM-PCA outperforms the snapshot POD. After $q = 3$, the EM-PCA is no more efficient than the the snapshot POD with the Lanczos algorithm, and its computational time is comparable to that of the snapshot POD. Finally, at $q = 4$ in Figure 15(d), the EM-PCA becomes slower than the snapshot POD. Therefore, the performance investigation results in Figure 15 convey that the EM-PCA is not much computationally favorable to other POD methods in contrary to its expected performance based on Table 1. Most of all, the differentials of computational time among the tested methods are too insignificant to determine the most efficient method.

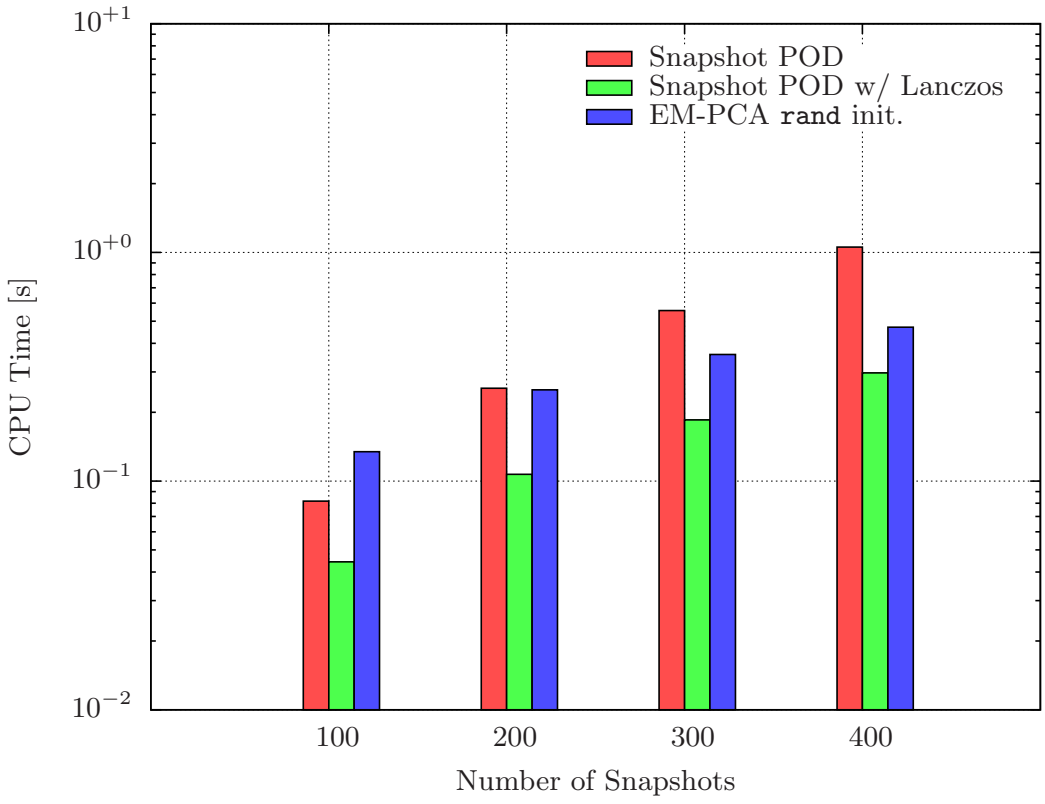


(a) $q = 1$

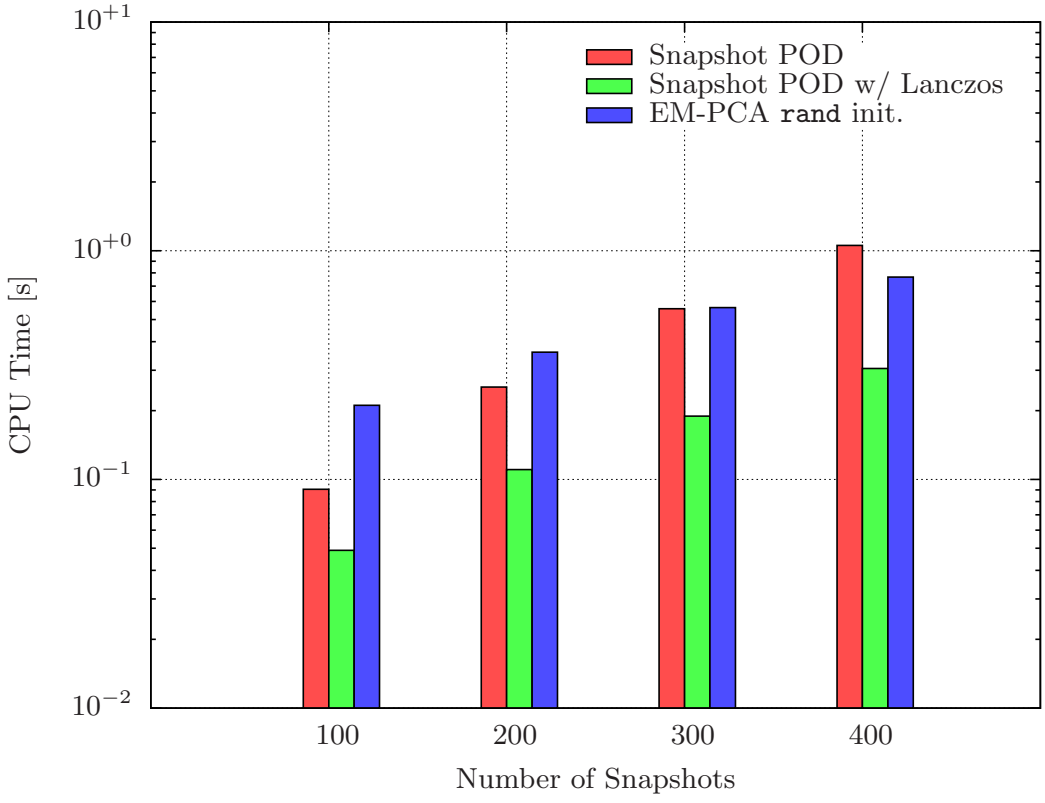


(b) $q = 2$

Figure 15: Variations in computational time with q increase for Euler airfoil pressure data



(c) $q = 3$



(d) $q = 4$

Figure 15: Variations in computational time with q increase for Euler airfoil pressure data

In order to delve into the observed ill-performance of the EM-PCA in Figure 15, this research examines the convergence characteristics of the EM-PCA. As an illustration, Figure 16 depicts the convergence history of the EM-PCA at $q = 7$, measured by a normalized RMSR of $\mathbf{W}$. As shown in Figure 16, the EM-PCA generally exhibits precipitative normalized RMSR reduction at early iterations, followed by a relatively slow convergence behavior throughout the rest of iterations. The convergence pattern in Figure 16 conveys that the EM-PCA struggles for convergence due to low-frequency errors that are hard to decay. Therefore, the EM-PCA necessitates more iterations, resulting in poor computational performance as noticed in Figure 15.

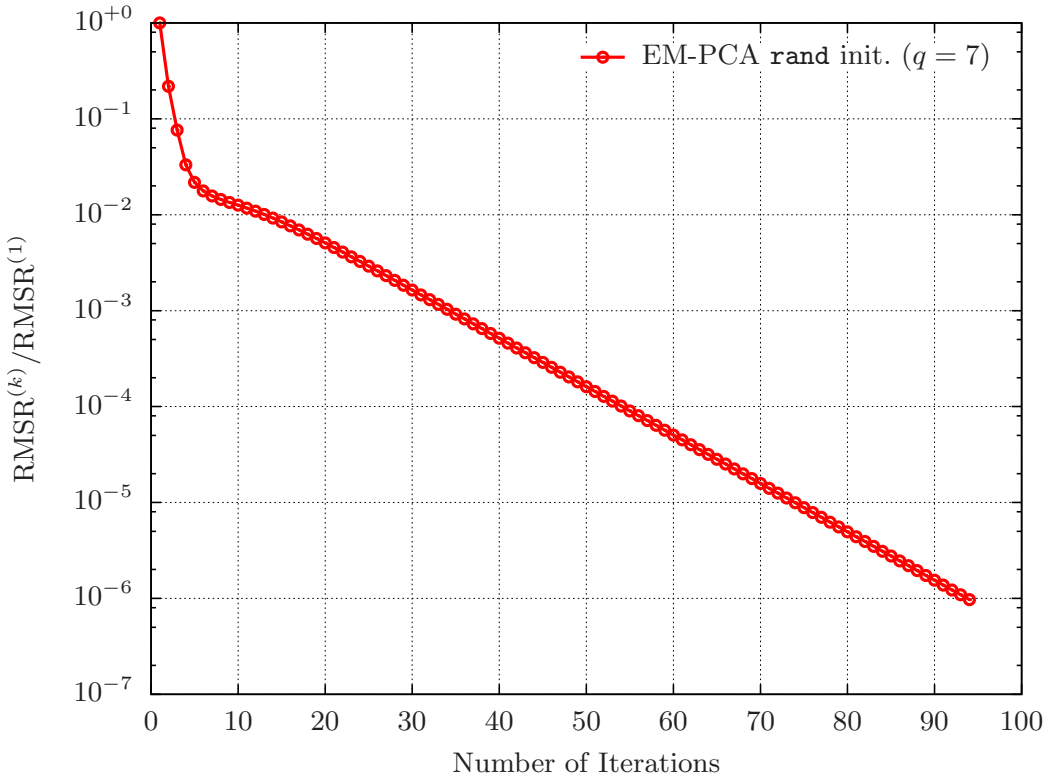


Figure 16: Convergence history of $\mathbf{W}$

CHAPTER IV

COMPARATIVE STUDY II: EM-PCA VS. GAPPY POD

4.1 Formulation of a Unifying Least-Squares Perspective

Unlike the transparent relationship between PPCA and the standard POD in Chapter 3, the analytical connection between the EM-PCA and gappy POD is obscure except that they both result from POD. Therefore, this section attempts to manipulate the formulations of the EM-PCA and gappy POD to reveal their theoretical relationship and to develop research questions and corresponding hypotheses associated with Research Objective 2 in Chapter 1.

4.1.1 Reformulation of Gappy POD and the EM-PCA

4.1.1.1 Gappy POD Recast in Forms of Matrix Multiplication

The least-squares problem for gappy POD in Eq. (10), translated from the gappy norm to the L^2 norm, is

$$\min. \quad \left\|\mathbf{n}_j \circ (\mathring{\mathbf{y}}_j - \mathbf{V}_q \mathbf{b}_j)\right\|_{L^2}^2 \quad \text{w.r.t.} \quad \mathbf{b}_j,$$

which is not as transparent as ordinary matrix multiplication due to the Hadamard product. In order to facilitate a comparative study, Lee and Mavris[32,33] proposes to recast the Hadamard product into matrix multiplication such that

$$\mathbf{n}_j \circ \mathbf{y}_j = \mathbf{N}_j \mathbf{y}_j \tag{26}$$

by introducing a diagonal matrix $\mathbf{N}_j \in \mathbb{R}^{d \times d}$ for $\mathbf{n}_j \in \mathbb{R}^d$, which lists $\mathbf{n}_j$ in its diagonal, i.e., $\mathbf{N}_j = \mathrm{diag}(\mathbf{n}_j)$. Note that $\mathbf{N}_j$ is symmetric, for it is diagonal, and because of zeros and ones in the diagonal, it is positive semi-definite and singular. Moreover, $\mathbf{N}_j$ is a projection since $\mathbf{N}_j^2 = \mathbf{N}_j$. With the help of the relationship in Eq. (26), the gappy norm defined in Eq. (7) can be expressed as

$$\|\mathbf{y}_j\|_n^2 = (\mathbf{y}_j, \mathbf{y}_j)_n = (\mathbf{n}_j \circ \mathring{\mathbf{y}}_j, \mathbf{n}_j \circ \mathring{\mathbf{y}}_j)_{L^2} = (\mathbf{N}_j \mathbf{y}_j, \mathbf{N}_j \mathbf{y}_j)_{L^2} = \mathbf{y}_j^{\mathrm{T}} \mathbf{N}_j \mathbf{y}_j, \tag{27}$$

which reveals that a squared gappy norm is equivalent to a weighted inner product of $\mathbf{y}_j$ with either zero or one weight.

With the transformation shown in Eq. (27), the squared estimation residual of gappy POD, shown in Eq. (9), can be rephrased in matrix multiplications:

$$r_j^2 = \|\mathbf{y}_j - \mathbf{V}_q\mathbf{b}_j\|_{\mathrm{n}}^2 = \left(\mathbf{n}_j \circ \left(\mathring{\mathbf{y}}_j - \mathbf{V}_q\mathbf{b}_j\right), \mathbf{n}_j \circ \left(\mathring{\mathbf{y}}_j - \mathbf{V}_q\mathbf{b}_j\right)\right)_{L^2}$$
$$= \left(\mathbf{N}_j\left(\mathbf{y}_j - \mathbf{V}_q\mathbf{b}_j\right), \mathbf{N}_j\left(\mathbf{y}_j - \mathbf{V}_q\mathbf{b}_j\right)\right)_{L^2} = \mathbf{y}_j^{\mathrm{T}}\mathbf{N}_j\mathbf{y}_j - 2\mathbf{y}_j^{\mathrm{T}}\mathbf{N}_j\mathbf{V}_q\mathbf{b}_j + \mathbf{b}_j^{\mathrm{T}}\left(\mathbf{V}_q^{\mathrm{T}}\mathbf{N}_j\mathbf{V}_q\right)\mathbf{b}_j.$$

As with the previous derivation process of gappy POD in Section 2.1.2, the stationary point of $\mathbf{b}_j$ can be found after taking a derivative of r_j^2 with respect to $\mathbf{b}_j$ and requiring it to vanish:

$$\left.\frac{\partial r_j^2}{\partial \mathbf{b}_j}\right|_{\mathbf{y}_j,\mathbf{V}_q} = -2\left(\mathbf{N}_j\mathbf{V}_q\right)^{\mathrm{T}}\mathbf{y}_j + 2\mathbf{b}_j^{\mathrm{T}}\left(\mathbf{V}_q^{\mathrm{T}}\mathbf{N}_j\mathbf{V}_q\right) = 0,$$

which reduces to a system of q linear equations such that

$$\left(\mathbf{V}_q^{\mathrm{T}}\mathbf{N}_j\mathbf{V}_q\right)\mathbf{b}_j = \left(\mathbf{N}_j\mathbf{V}_q\right)^{\mathrm{T}}\mathbf{y}_j.$$

Finally, the optimal coefficient $\mathbf{b}_j$ is determined by

$$\mathbf{b}_j = \left(\mathbf{V}_q^{\mathrm{T}}\mathbf{N}_j\mathbf{V}_q\right)^{-1}\left(\mathbf{N}_j\mathbf{V}_q\right)^{\mathrm{T}}\mathbf{y}_j, \tag{28}$$

which corresponds to b_{ij} in Eq. (11) derived in Section 2.1.2. Note that the matrix multiplied by $\mathbf{y}_j$ in Eq. (28), i.e., $(\mathbf{V}_q^{\mathrm{T}}\mathbf{N}_j\mathbf{V}_q)^{-1}(\mathbf{N}_j\mathbf{V}_q)^{\mathrm{T}}$, must change in accordance with $\mathring{\mathbf{y}}_j$ because $\mathbf{N}_j$ is unique to each $\mathring{\mathbf{y}}_j$. As a result, gappy POD requires as many evaluations of $(\mathbf{V}_q^{\mathrm{T}}\mathbf{N}_j\mathbf{V}_q)^{-1}(\mathbf{N}_j\mathbf{V}_q)^{\mathrm{T}}$ as the number of $\mathring{\mathbf{y}}_j$. Note that despite more transparency, Eq. (28) is computationally less efficient than the original formulation in Eq. (11) because of matrix operations including a full matrix $\mathbf{N}_j$. Therefore, gappy POD is implemented based on its original formulation described in Section 2.1.2.

4.1.1.2 EM-PCA as an Iterative Optimizer

Unlike gappy POD, which directly tackles a least-squares problem in a deterministic way, the EM-PCA does not address an explicit form of a least-squares problem; instead, the EM-PCA is designed to maximize the expected log-likelihood in Eq. (19) by alternating the E-step and the M-step to find probability parameter estimates. During iterations, the EM-PCA repeats both E- and M-steps in such a way that the E-step computes unknown variables

while keeping parameter estimates fixed and, similarly, the subsequent M-step evaluates the parameter estimates while holding the unknown variables constant. Interestingly, this EM-PCA process is equivalent to iteratively solving a least-squares problem because the EM algorithm belongs to bound optimization methods that are known to carry out fixed-point iterations for optimization.[20] After all, provided that observations are free of measurement errors, the EM-PCA in Eq. (22) is actually identical to minimizing an averaged squared residual R^2 defined as

$$R^2 = \frac{1}{N}\sum_{j=1}^{N} \|\mathbf{y}_j - \mathbf{W}\mathbf{x}_j\|_{L^2}^2 = \frac{1}{N}\,\mathrm{tr}\left(\|\mathbf{Y} - \mathbf{W}\mathbf{X}\|_{L^2}^2\right) \tag{29}$$

in a fixed-point iteration fashion. Indeed, the same equations as those of the EM-PCA, listed in Eq. (22), can be achieved after R^2 in Eq. (29) is differentiated with respect to each variable, and then the first derivatives are equated to zero as follows:

$$\begin{aligned}
\left.\frac{\partial R^2}{\partial \mathbf{X}}\right|_{\mathbf{Y},\mathbf{W}} = 0 \quad &\Longrightarrow \quad \mathbf{X} = \left(\mathbf{W}^{\mathrm{T}}\mathbf{W}\right)^{-1}\mathbf{W}^{\mathrm{T}}\mathbf{Y}, \\
\left.\frac{\partial R^2}{\partial \mathbf{Y}}\right|_{\mathbf{X},\mathbf{W}} = 0 \quad &\Longrightarrow \quad \mathbf{Y} = \mathbf{W}\mathbf{X}, \\
\left.\frac{\partial R^2}{\partial \mathbf{W}}\right|_{\mathbf{X},\mathbf{Y}} = 0 \quad &\Longrightarrow \quad \mathbf{W} = \mathbf{Y}\mathbf{X}^{\mathrm{T}}\left(\mathbf{X}\mathbf{X}^{\mathrm{T}}\right)^{-1},
\end{aligned}$$

which ascertains that the EM-PCA implicitly minimizes R^2 in Eq. (29) in an iterative manner. Hence, similar to the least-squares problem of gappy POD in Eq. (10), the EM-PCA solves a de facto least-squares problem such that

$$\text{min.} \quad \left\|\mathbf{y}_j - \mathbf{W}\mathbf{x}_j\right\|_{L^2}^2 \quad \text{w.r.t.} \quad \mathbf{x}_j \tag{30}$$

to restore missing data in an observed variable $\mathbf{y}_j$. For a coefficient $\mathbf{x}_j$, a squared residual r_j^2 is evaluated as

$$r_j^2 = \|\mathbf{y}_j - \mathbf{W}\mathbf{x}_j\|_{L^2}^2 = \left(\mathbf{y}_j - \mathbf{W}\mathbf{x}_j, \mathbf{y}_j - \mathbf{W}\mathbf{x}_j\right)_{L^2} = \mathbf{y}_j^{\mathrm{T}}\mathbf{y}_j - 2\mathbf{y}_j^{\mathrm{T}}\mathbf{W}\mathbf{x}_j + \mathbf{x}_j^{\mathrm{T}}\left(\mathbf{W}^{\mathrm{T}}\mathbf{W}\right)\mathbf{x}_j,$$

and like the earlier derivation of $\mathbf{b}_j$ for gappy POD in Section 4.1.1.1, the optimal least-squares coefficient $\mathbf{x}_j$ of the EM-PCA can be found by the following stationary equation:

$$\left.\frac{\partial r_j^2}{\partial \mathbf{x}_j}\right|_{\mathbf{y}_j,\mathbf{W}} = -2(\mathbf{y}_j^{\mathrm{T}}\mathbf{W}) + 2\left(\mathbf{W}^{\mathrm{T}}\mathbf{W}\right)\mathbf{x}_j = 0,$$

which yields a coefficient $\mathbf{x}_j$ as

$$\mathbf{x}_j = \left(\mathbf{W}^{\mathrm{T}}\mathbf{W}\right)^{-1}\mathbf{W}^{\mathrm{T}}\mathbf{y}_j, \tag{31}$$

corresponding to $\langle \mathbf{x}_j \rangle$ of the E-step, shown in Eq. (22a). Note that $\mathbf{x}_j$ in Eq. (31), as opposed to $\mathbf{b}_j$ in Eq. (28), has a constant multiplying matrix $(\mathbf{W}^{\mathrm{T}}\mathbf{W})^{-1}\mathbf{W}^{\mathrm{T}}$ by $\mathbf{y}_j$, regardless of a gappy snapshot $\mathring{\mathbf{y}}_j$. Thus, the least-squares coefficient evaluation of the EM-PCA is expected to be more efficient than that of gappy POD per evaluation.

4.1.2 Algorithmic Analysis of Gappy POD and the EM-PCA

4.1.2.1 Generalized least-squares Problem Integrating Gappy POD and the EM-PCA

As elucidated previously in Section 4.1.1, gappy POD solves a least-squares problem explicitly, and interestingly, the EM-PCA does so implicitly. For a methodical comparison of both gappy POD and the EM-PCA, each least-squares problem of gappy POD and the EM-PCA in Eq. (10) and Eq. (30), respectively, can be generalized as

$$\text{min.} \quad \left\|\mathbf{y}_j - \mathbf{\Phi}\mathbf{c}_j\right\|_\alpha^2 \quad \text{w.r.t.} \quad \mathbf{\Phi} \text{ and } \mathbf{c}_j, \tag{32}$$

where $\mathbf{\Phi}$ is a basis, α is a norm, and $\mathbf{c}_j$ is a coefficient for $\mathbf{\Phi}$. Therefore, in view of a least-squares perspective, the two missing-data reconstruction methods share the generalized least-squares problem in Eq. (32), which requires the evaluation of a basis $\mathbf{\Phi}$ and a least-squares coefficient $\mathbf{c}_j$. Their difference, however, lies in their choices of a basis $\mathbf{\Phi}$ for a subspace projection and a norm α for a squared residual evaluation, both of which together result in a disparate coefficient $\mathbf{c}_j$ that reflects their algorithmic characteristics to address missing data estimation.

Overall, Table 2 summarizes the similarities and the disparities of gappy POD and the EM-PCA to contrast each step of both formulations. To begin with the similarities, they both solve a least-squares problem that reduces to a twofold algorithm: basis and least-squares coefficient evaluations. Despite their similar processes, they differ in each step due to their disparities in a basis $\mathbf{\Phi}$ and a norm α. In detail, to evaluate a basis, gappy POD exploits POD for a POD basis $\mathbf{V}_q$, which is orthogonal, whereas the EM-PCA relies on its M-step for a factor-loading matrix $\mathbf{W}$, which is non-orthogonal. Likewise, to

Table 2: Least-squares formulations of gappy POD and the EM-PCA

	gappy POD: $\mathbf{V}_q + \|\cdot\|_n$	EM-PCA: $\mathbf{W} + \|\cdot\|_{L^2}$
problem	min. $\left\|\mathbf{y}_j - \mathbf{V}_q\mathbf{b}_j\right\|_n^2$ w.r.t. $\mathbf{b}_j$	min. $\left\|\mathbf{y}_j - \mathbf{W}\mathbf{x}_j\right\|_{L^2}^2$ w.r.t. $\mathbf{x}_j$ and $\mathbf{W}$
approximation	$(\mathbf{N}_j\mathbf{V}_q)\,\mathbf{b}_j = \mathbf{N}_j\mathbf{y}_j$	$\mathbf{W}\mathbf{x}_j = \mathbf{y}_j$
normal equation	$\left(\mathbf{V}_q^{\mathrm{T}}\mathbf{N}_j\mathbf{V}_q\right)\mathbf{b}_j = (\mathbf{N}_j\mathbf{V}_q)^{\mathrm{T}}\mathbf{y}_j$	$\left(\mathbf{W}^{\mathrm{T}}\mathbf{W}\right)\mathbf{x}_j = \mathbf{W}^{\mathrm{T}}\mathbf{y}_j$
projection	$(\mathbf{N}_j\mathbf{V}_q)\left(\mathbf{V}_q^{\mathrm{T}}\mathbf{N}_j\mathbf{V}_q\right)^{-1}(\mathbf{N}_j\mathbf{V}_q)^{\mathrm{T}}$	$\mathbf{W}\left(\mathbf{W}^{\mathrm{T}}\mathbf{W}\right)^{-1}\mathbf{W}^{\mathrm{T}}$
coefficient	$\mathbf{b}_j = \left(\mathbf{V}_q^{\mathrm{T}}\mathbf{N}_j\mathbf{V}_q\right)^{-1}(\mathbf{N}_j\mathbf{V}_q)^{\mathrm{T}}\mathbf{y}_j$	$\mathbf{x}_j = \left(\mathbf{W}^{\mathrm{T}}\mathbf{W}\right)^{-1}\mathbf{W}^{\mathrm{T}}\mathbf{y}_j$
basis	$\mathbf{V}_q$ by POD	$\mathbf{W} = \mathbf{Y}\mathbf{X}^{\mathrm{T}}\left(\mathbf{X}\mathbf{X}^{\mathrm{T}}\right)^{-1}$

evaluate a least-squares coefficient, owing to their norm difference, both gappy POD and the EM-PCA end up with dissimilar least square problems: the former being a weighted least-squares problem and the latter being an ordinary least-squares problem.

Table 2 conveys that each step of the EM-PCA is more efficient than that of gappy POD for the following reasons. First, regarding a basis evaluation, gappy POD relies on a POD operation that involves numerically expensive EVD or SVD. In contrast, the EM-PCA generates a basis through mere matrix multiplication and inversion, both of which are less demanding than either EVD or SVD. The computational cost of $\mathbf{W}$ is expected to grow with q because of the q-by-q matrix inversion, $(\mathbf{X}\mathbf{X}^{\mathrm{T}})^{-1}$, but that of $\mathbf{V}_q$ is expensive cubically to N or d, depending on snapshot or standard POD, respectively. Second, as to a coefficient evaluation, the computational advantage of the EM-PCA over gappy POD becomes more conspicuous. Since $\mathbf{N}_j$ is unique to $\mathbf{y}_j$, the matrix multiplied by $\mathbf{y}_j$, $(\mathbf{V}_q^{\mathrm{T}}\mathbf{N}_j\mathbf{V}_q)^{-1}(\mathbf{N}_j\mathbf{V}_q)^{\mathrm{T}}$, necessitates a new evaluation for every data-missing snapshot. Therefore, gappy POD must compute the multiplied matrix the same number of times as the number of data-missing snapshots. Consequently, the coefficient evaluation of gappy POD is susceptible to a data set that contains a number of data-missing snapshots due to scattered missing data. However, the EM-PCA utilizes a constant multiplied matrix by $\mathbf{y}_j$, $(\mathbf{W}^{\mathrm{T}}\mathbf{W})^{-1}\mathbf{W}^{\mathrm{T}}$, invariant to $\mathbf{y}_j$. Thus, the EM-PCA requires a single evaluation of the multiplied matrix regardless of the number of data-missing snapshots. As a result, the theoretical dissection of gappy POD

Table 3: Least-squares formulations of hybrid algorithms

	Hybrid 1: $\mathbf{W} + \|\cdot\|_n$	Hybrid 2: $\mathbf{V}_q + \|\cdot\|_{L^2}$
problem	min. $\left\|\mathbf{y}_j - \mathbf{W}\mathbf{b}_j\right\|_n^2$ w.r.t. $\mathbf{b}_j$	min. $\left\|\mathbf{y}_j - \mathbf{V}_q\mathbf{x}_j\right\|_{L^2}^2$ w.r.t. $\mathbf{x}_j$
approximation	$(\mathbf{N}_j\mathbf{W})\,\mathbf{b}_j = \mathbf{N}_j\mathbf{y}_j$	$\mathbf{V}_q\mathbf{x}_j = \mathbf{y}_j$
normal equation	$\left(\mathbf{W}^{\mathrm{T}}\mathbf{N}_j\mathbf{W}\right)\mathbf{b}_j = (\mathbf{N}_j\mathbf{W})^{\mathrm{T}}\mathbf{y}_j$	$\left(\mathbf{V}_q^{\mathrm{T}}\mathbf{V}_q\right)\mathbf{x}_j = \mathbf{V}_q^{\mathrm{T}}\mathbf{y}_j$
projection	$(\mathbf{N}_j\mathbf{W})\left(\mathbf{W}^{\mathrm{T}}\mathbf{N}_j\mathbf{W}\right)^{-1}(\mathbf{N}_j\mathbf{W})^{\mathrm{T}}$	$\mathbf{V}_q\left(\mathbf{V}_q^{\mathrm{T}}\mathbf{V}_q\right)^{-1}\mathbf{V}_q^{\mathrm{T}}$
coefficient	$\mathbf{b}_j = \left(\mathbf{W}^{\mathrm{T}}\mathbf{N}_j\mathbf{W}\right)^{-1}(\mathbf{N}_j\mathbf{W})^{\mathrm{T}}\mathbf{y}_j$	$\mathbf{x}_j = \left(\mathbf{V}_q^{\mathrm{T}}\mathbf{V}_q\right)^{-1}\mathbf{V}_q^{\mathrm{T}}\mathbf{y}_j$
basis	$\mathbf{W} = \mathbf{Y}\mathbf{B}^{\mathrm{T}}\left(\mathbf{B}\mathbf{B}^{\mathrm{T}}\right)^{-1}$	$\mathbf{V}_q$ given by POD

and the EM-PCA, recapitulated in Table 2, infers that the EM-PCA is preferable to gappy POD in concern of computational cost for a given fixed number of basis vectors.

4.1.2.2 Hybrid Algorithms of Gappy POD and the EM-PCA

Since a basis and a norm are the two fundamental elements characterizing both gappy POD and the EM-PCA, their hybrid algorithms can easily be devised by blending their bases and norms. For example, one algorithm can be composed of a non-orthogonal basis $\mathbf{W}$ and the gappy norm, each of which derives from the EM-PCA and gappy POD, respectively. Likewise, the other one can be comprised of an orthogonal basis $\mathbf{V}_q$ and the L^2 norm, each of which is adopted from gappy POD and the EM-PCA, respectively. For notational convenience, the former and the latter hybrid algorithms are termed as "Hybrid 1" and "Hybrid 2," respectively, whose formulation characteristics are delineated in Table 3, similar to Table 2. These hybrid algorithms are particularly useful for investigating different basis and norm effects on missing data estimation in comparison with their originals later in Section 4.2. Note that the formulation of Hybrid 1 can analytically derive $\mathbf{b}_j$, but it cannot do so for $\mathbf{W}$ because of the singularity of $\mathbf{N}_j$. Thus, $\mathbf{W}$ of Hybrid 1 in Table 3 is a carbon copy of $\mathbf{W}$ in the gappy POD formulation for the construction of Hybrid 1. Since Hybrid 1 is an improvised formulation in this sense, it is sometimes numerically insecure in evaluating the matrix inverse operation of $\mathbf{W}$.

4.1.3 Further Development of Research Questions and Hypotheses

After gappy POD is reformulated and the EM-PCA is construed as an iterative optimizer, both gappy POD and the EM-PCA are found to be least-squares methods. Based on their algorithmic resemblance, Methodological Hypothesis 2, related to Research Objective 2, can be constructed.

Methodological Hypothesis 2. A unifying least-squares perspective integrates both the EM-PCA and gappy POD within a common formulation framework.

Since the antithetically formulated EM-PCA and gappy POD can be juxtaposed as shown in Table 2, the formulation disparities between the EM-PCA and gappy POD raise a subsequent research question as follows:

Research Question 2.1. What are the effects of the disparate bases and norms on estimation error reduction and the computational performance of the EM-PCA and gappy POD?

Due to insufficient knowledge regarding the ramifications of the different bases and norms, this research is unable to properly conjecture the effects of the different bases and norms. Therefore, this research attempts to conduct a series of experiments in the following sections to collect enough observations to build a hypothesis corresponding to Research Question 2.1. In the experiments, the theoretical effect on error reduction will be measured in terms of a root mean square error (RMSE), and similarly, the numerical effect on performance will be accessed in terms of the CPU time and the total iteration number.

4.2 Qualitative Investigation of Different Basis and Norm Effects

In order to examine the intrinsic basis and norm differences revealed in Section 4.1, this research starts by delving into the theoretical aspects of the bases and norms. To begin with, regarding the basis difference, which determines the quality of a subspace projection, gappy POD exploits an orthogonal basis $\mathbf{V}_q$ whereas the EM-PCA employs a non-orthogonal basis $\mathbf{W}$. Since an orthogonal basis is known to produce the lowest projection error than any other linear basis at a given number of basis vectors,[18] $\mathbf{V}_q$ is lucidly preferable to $\mathbf{W}$.

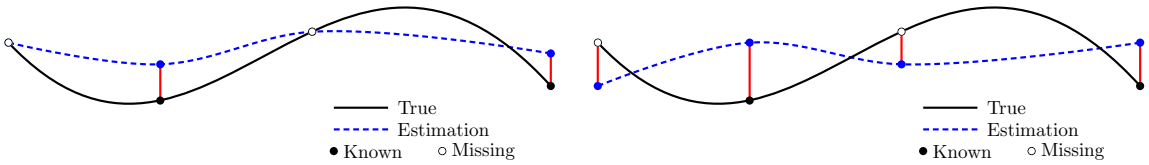
Hence, provided that both $\mathbf{V}_q$ and $\mathbf{W}$ have the same number of basis vectors, the former yields fewer estimation errors than the latter. Notwithstanding the desirable orthogonality of $\mathbf{V}_q$, in iterations, gappy POD does not use the true $\mathbf{V}_q$ but instead an estimated $\widetilde{\mathbf{V}}_q$ obtained from a yet-to-be-converged data set $\widetilde{\mathbf{Y}}$. Similar to gappy POD, the EM-PCA uses an estimated $\widetilde{\mathbf{W}}$ evaluated with an intermediate snapshot ensemble $\widetilde{\mathbf{Y}}$ as a substitute for the true $\mathbf{W}$. Therefore, the aforementioned analytical characteristic of $\mathbf{V}_q$ compared to that of $\mathbf{W}$ does not hold in determining which basis is superior to the other during actual iterations. In general, the property of $\mathbf{V}_q$ and $\mathbf{W}$ for a subspace projection could be directly carried over to $\widetilde{\mathbf{V}}_q$ and $\widetilde{\mathbf{W}}$ only when $\widetilde{\mathbf{Y}}$ in iterations would be very close to $\mathbf{Y}$. If so, $\widetilde{\mathbf{V}}_q$ would be better than $\widetilde{\mathbf{W}}$ at reducing estimation residuals; otherwise, $\widetilde{\mathbf{V}}_q$ would be merely as good as $\widetilde{\mathbf{W}}$. Since the convergence of $\widetilde{\mathbf{Y}}$ hinges on several factors, such as the amount and spread of missing data as well as the inherent nonlinearity of the data, the quality of approximate bases $\widetilde{\mathbf{V}}_q$ and $\widetilde{\mathbf{W}}$ will be highly affected by those factors.

Next, regarding the norm difference, which affects the evaluation of an estimation residual, gappy POD utilizes the gappy norm whereas the EM-PCA uses the L^2 norm. Because of their norm difference, both methods end up with dissimilar least-squares problems: a weighted least-squares problem for gappy POD and an ordinary least-squares problem for the EM-PCA. In particular, the dissimilarity between their least-squares problems can be captured by their projection matrices:

$$\mathbf{P}_{\text{GPOD}} = (\mathbf{N}_j\mathbf{V}_q)\left(\mathbf{V}_q^{\mathrm{T}}\mathbf{N}_j\mathbf{V}_q\right)^{-1}(\mathbf{N}_j\mathbf{V}_q)^{\mathrm{T}}, \tag{33a}$$

$$\mathbf{P}_{\text{EM-PCA}} = \mathbf{W}\left(\mathbf{W}^{\mathrm{T}}\mathbf{W}\right)^{-1}\mathbf{W}^{\mathrm{T}}, \tag{33b}$$

where $\mathbf{P}_{\text{GPOD}}$ in Eq. (33a) is the projection of gappy POD and $\mathbf{P}_{\text{EM-PCA}}$ in Eq. (33b) is that of the EM-PCA. Both projections are similar in that they are orthogonal projections on inner product spaces defined by the gappy and L^2 norms; however, they are dissimilar in the way they assess estimation residuals. As illustrated in Figure 17(a), due to $\mathbf{N}_j$, induced by the gappy norm, $\mathbf{P}_{\text{GPOD}}$ neglects unknowns, so it is rigorous in computing r_j^2 by tracing variations in only known data; thus, a new estimation always anchors to locations of missing data, which is more like *interpolation*. By contrast, as delineated in Figure 17(b), because



(a) Gappy norm (b) L^2 norm

Figure 17: The evaluation of an estimation residual implied by a norm

of the L^2 norm, $\mathbf{P}_{\text{EM-PCA}}$ treats data equally without regard to their availability, so it is less stringent in evaluating r_j^2; therefore, a new estimation does not necessarily pass through any points, which is similar to *regression*.

For instance, suppose $\mathbf{y}_j \in \mathbb{R}^4$ and its corresponding mask vector $\mathbf{n}_j = (0,1,0,1)^{\mathrm{T}}$, denoting missing data at the first and third elements of $\mathbf{y}_j$. The estimation residual at the $(k+1)^{\text{th}}$ iteration is calculated with the gappy norm as shown in Eq. (34a), corresponding to Figure 17(a). On the other hand, the same $(k+1)^{\text{th}}$ estimation residual is computed with the L^2 norm as shown in Eq. (34b), representing Figure 17(b). After all, different norms theoretically imply dissimilar mechanisms of missing-data estimation, which would not necessarily convey which norm would generate better estimation results.

$$r_j^{(k+1)} = \left\| \widetilde{\mathbf{y}}_j^{(k+1)} - \widetilde{\mathbf{y}}_j^{(k)} \right\|_{\mathrm{n}} = \begin{bmatrix} 0 \\ \widetilde{y}_{2j}^{(k+1)} \\ 0 \\ \widetilde{y}_{4j}^{(k+1)} \end{bmatrix} - \begin{bmatrix} 0 \\ y_{2j}^{(k)} \\ 0 \\ y_{4j}^{(k)} \end{bmatrix} = \begin{bmatrix} 0 \\ \widetilde{y}_{2j}^{(k+1)} - y_{2j}^{(k)} \\ 0 \\ \widetilde{y}_{4j}^{(k+1)} - y_{4j}^{(k)} \end{bmatrix} \tag{34a}$$

$$r_j^{(k+1)} = \left\| \widetilde{\mathbf{y}}_j^{(k+1)} - \widetilde{\mathbf{y}}_j^{(k)} \right\|_{L^2} = \begin{bmatrix} \widetilde{y}_{1j}^{(k+1)} \\ \widetilde{y}_{2j}^{(k+1)} \\ \widetilde{y}_{3j}^{(k+1)} \\ \widetilde{y}_{4j}^{(k+1)} \end{bmatrix} - \begin{bmatrix} \widetilde{y}_{1j}^{(k)} \\ y_{2j}^{(k)} \\ \widetilde{y}_{3j}^{(k)} \\ y_{4j}^{(k)} \end{bmatrix} = \begin{bmatrix} \widetilde{y}_{1j}^{(k+1)} - \widetilde{y}_{1j}^{(k)} \\ \widetilde{y}_{2j}^{(k+1)} - y_{2j}^{(k)} \\ \widetilde{y}_{3j}^{(k+1)} - \widetilde{y}_{3j}^{(k)} \\ \widetilde{y}_{4j}^{(k+1)} - y_{4j}^{(k)} \end{bmatrix} \tag{34b}$$

Table 4: Algorithmic comparison to isolate each basis and norm effect

comparison			common	implementation pair to compare					
$\mathbf{V}_q$	vs.	$\mathbf{W}$	$\|\cdot\|_n$	gappy POD:	$\mathbf{V}_q + \|\cdot\|_n$	&	Hybrid 1:	$\mathbf{W} + \|\cdot\|_n$
			$\|\cdot\|_{L^2}$	EM-PCA:	$\mathbf{W} + \|\cdot\|_{L^2}$	&	Hybrid 2:	$\mathbf{V}_q + \|\cdot\|_{L^2}$
$\|\cdot\|_n$	vs.	$\|\cdot\|_{L^2}$	$\mathbf{V}_q$	gappy POD:	$\mathbf{V}_q + \|\cdot\|_n$	&	Hybrid 2:	$\mathbf{V}_q + \|\cdot\|_{L^2}$
			$\mathbf{W}$	EM-PCA:	$\mathbf{W} + \|\cdot\|_{L^2}$	&	Hybrid 1:	$\mathbf{W} + \|\cdot\|_n$

4.3 Quantitative Investigation of Different Basis and Norm Effects

Although the dissimilar bases and norms of gappy POD and the EM-PCA are examined from a theoretical viewpoint in Section 4.2, the previous analysis was insufficient for determining which basis or norm is superior to the other in practice. Therefore, the earlier qualitative analyses in Section 4.2 necessitate supplementary quantitative investigations. For this purpose, Lee and Mavris[30] devised a series of numerical experiments using two types of missing data structures that generalize the applications of gappy POD in the literature. In order to measure different basis and norm effects, they compared the RMSE histories of both gappy POD and the EM-PCA to those of their two hybrid algorithms.

4.3.1 Comparison Strategy to Isolate Different Basis and Norm Effects

The hybrid algorithms, formulated in Section 4.1.2.2, facilitate the quantitative study on the effects of the basis and norm differences. Because the direct comparison of gappy POD and the EM-PCA results in compound effects due to mixed bases and norms, the contributions of the bases and norms cannot be properly retrieved. However, as summarized in Table 4, comparisons of the original and hybrid algorithms can separately illustrate the effects of the different bases and norms. For example, the comparison of gappy POD and Hybrid 1 captures the basis difference effect at the same gappy norm, and so does the comparison of the EM-PCA and Hybrid 2 at the same L^2 norm. Similarly, the comparison of gappy POD and Hybrid 2 delineates the norm difference effect under the same $\mathbf{V}_q$, and so does the comparison of the EM-PCA and Hybrid 1 under the same $\mathbf{W}$. By virtue of the two hybrid methods, each basis and norm effect can be effectively isolated through the systematic

comparisons, summarized in Table 4.

4.3.2 Implementation of the Algorithms

In order to quantitatively dissect the effects of bases and norms, Lee and Mavris[30] implemented four algorithms—gappy POD, the EM-PCA, and their two hybrids—with three variations as follows:

(i) Whether to keep a sample mean constant in iterations.

Implementations that hold a sample mean are denoted by appending "$\boldsymbol{\mu}$ inv." to their name such as "EM-PCA $\boldsymbol{\mu}$ inv.," and those that do not are indicated by adding "$\boldsymbol{\mu}$ var." to the end of their name such as "EM-PCA $\boldsymbol{\mu}$ var."

(ii) The way to initialize a factor-loading $\mathbf{W}$.

Among the four algorithms, the EM-PCA and Hybrid 1 can initialize a factor-loading $\mathbf{W}$ in either a random or informed manner. For clarity, in the names of both EM-PCA and Hybrid 1 implementations, those initializing $\mathbf{W}$ with a random matrix are represented by "`rand`," and those initializing $\mathbf{W}$ with an estimated POD basis like gappy POD are indicated with "$\mathbf{V}_e$": the POD basis $\mathbf{V}_q^{(0)}$ obtained from $\widetilde{\mathbf{Y}}^{(0)}$ whose missing data are filled with a sample mean before the onset of iterations. Because of $\mathbf{V}_e$, the implementations of both EM-PCA and Hybrid 1 can have the same basis initialization as gappy POD and Hybrid 2, which is conducive to unbiased comparative studies.

(iii) The way to evaluate a POD basis $\mathbf{V}_q$.

The two algorithms employing a POD basis, i.e., gappy POD and Hybrid 2, can expedite POD by capitalizing on the Lanczos algorithm in lieu of either EVD or SVD. In the names of both gappy POD and Hybrid 2 implementations, those that benefit from the Lanczos algorithm are specified with "Lanczos" such as "GPOD $\boldsymbol{\mu}$ inv.: Lanczos." For this research, the Lanczos algorithm is realized with a MATLAB function `eigs`, which internally invokes the Fortran Library ARPACK.[38] Note that implementations using the Lanczos algorithm accelerate only a basis evaluation step,

Table 5: Sample-mean invariant implementations

Implementation name	Algorithm	Norm	Basis:	initialization	evaluation
EM-PCA $\boldsymbol{\mu}$ inv.: `rand`	EM-PCA	$\|\cdot\|_{L^2}$	$\mathbf{W}$	$\mathbf{W}^{(0)} = \texttt{rand}$	M-step
EM-PCA $\boldsymbol{\mu}$ inv.: $\mathbf{V}_e$				$\mathbf{W}^{(0)} = \mathbf{V}_e$	M-step
GPOD $\boldsymbol{\mu}$ inv.	gappy POD	$\|\cdot\|_n$	$\mathbf{V}_q$	$\mathbf{V}_q^{(0)} = \mathbf{V}_e$	SVD
GPOD $\boldsymbol{\mu}$ inv.: Lanczos					Lanczos
Hybrid1 $\boldsymbol{\mu}$ inv.: `rand`	Hybrid 1	$\|\cdot\|_n$	$\mathbf{W}$	$\mathbf{W}^{(0)} = \texttt{rand}$	M-step
Hybrid1 $\boldsymbol{\mu}$ inv.: $\mathbf{V}_e$				$\mathbf{W}^{(0)} = \mathbf{V}_e$	M-step
Hybrid2 $\boldsymbol{\mu}$ inv.	Hybrid 2	$\|\cdot\|_{L^2}$	$\mathbf{V}_q$	$\mathbf{V}_q^{(0)} = \mathbf{V}_e$	SVD
Hybrid2 $\boldsymbol{\mu}$ inv.: Lanczos					Lanczos

producing identical results to those without using it. Therefore, implementations with the Lanczos algorithm will be used only for computation performance investigations in Section 4.4 later on.

In summary, Table 5 delineates the implementations of the four algorithms, keeping a sample mean invariant during the iterations. The other half of the implementations, allowing for changes in the sample mean, have exactly the same structures, shown in Table 5.

For convergence monitoring, all the implementations inspect the normalized RMSR of an estimated snapshot $\widetilde{\mathbf{y}}_j$ and the number of iterations to see if either of their thresholds is violated as follows:

$$\frac{\text{RMSR}^{(k)}}{\text{RMSR}^{(1)}} < 10^{-6} \quad \text{or} \quad \text{the number of iterations} > 10^4, \tag{35}$$

where

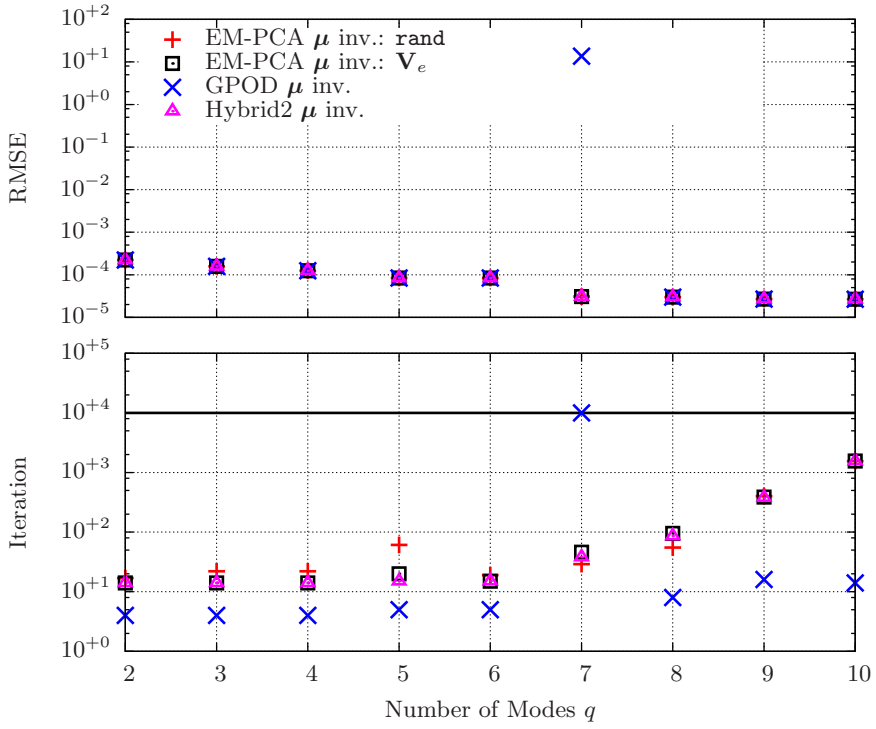
$$\text{RMSR}^{(k)} = \sqrt{\frac{1}{dN}\sum_{j=1}^{N}\left\|\widetilde{\mathbf{y}}_j^{(k)} - \widetilde{\mathbf{y}}_j^{(k-1)}\right\|_{L^2}^2}.$$

To measure each basis and norm effect, this research also calculates an RMSE of an estimated snapshot $\widetilde{\mathbf{y}}_j$, defined similarly to an RMSR, by evaluating differences between the true and estimated values. All numerical experiments herein were carried out in a MATLAB R2007b environment on an Intel Pentium dual-core 2.8 GHZ processor with 1 GB memory, and MATLAB `tic` and `toc` functions were utilized to record the computational time of all

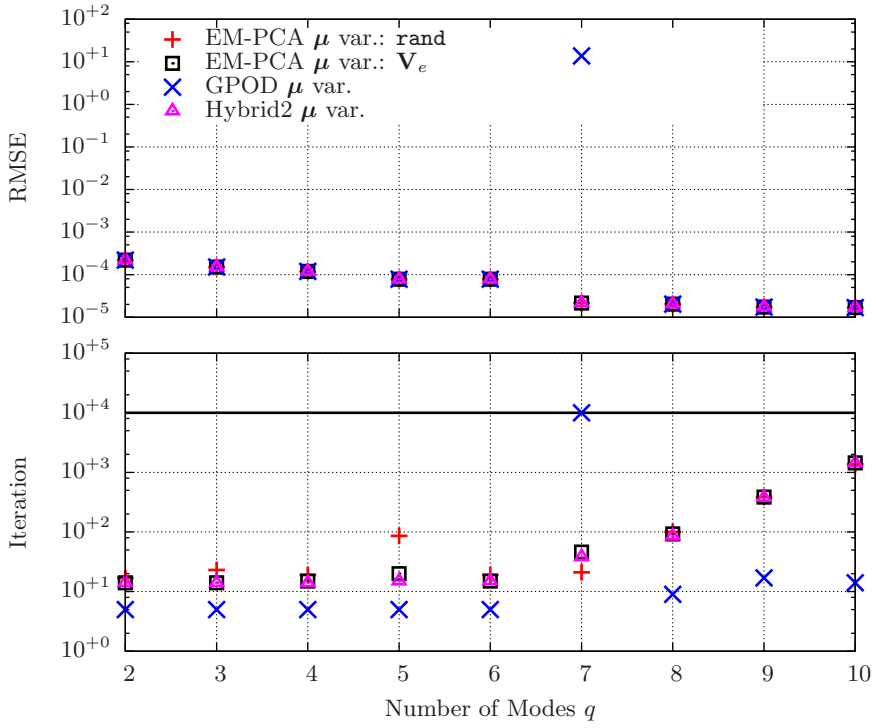
the implementations. Note that algorithms that take a random initialization for $\mathbf{W}$, i.e., the EM-PCA and Hybrid 1, were run 100 times consecutively to neutralize the effect of randomness.

4.3.3 Sample Data Generation

For the numerical investigation, this research compiled steady-state pressure coefficient C_p data from the Euler CFD flow analysis of the RAE 2822 airfoil using a generic numerical compressible airflow solver (GENCAS).[54–56] The CFD flow solver was set to employ the following numerical techniques: (i) Roe's flux difference splitting (FDS) along with second-order monotone upstream-centered schemes for conservation laws (MUSCL) reconstruction with a minmod limiter; and (ii) implicit lower-upper symmetric Gauss-Seidel (LU-SGS) time-stepping. With the help of the GENCAS, a 4257-by-100 C_p data set was obtained from a total of 100 analysis snapshots with variations in a Mach number from 0.6 to 0.8 and an angle of attack from 1° to 3°. For strategic parameter space exploration, the combinations of the two parameters are produced by JMP software[24] based on a maximum-entropy space-filling design, one of the DoE for computer simulations. Once the data set is obtained, this research attempted to randomly discard 30% of the data in order to examine the basis and norm effects, producing two artificially-marred data sets: (i) the first sample data set, missing 29.9507% of data only in the randomly-chosen 57^{th} snapshot, and (ii) the second sample data set, missing 29.9746% of the data spread over the entire snapshot ensemble. These two incomplete data sets exemplify the applications of gappy POD in the literature; for example, the first sample data set epitomizes such applications as flow data assimilation[84] and inverse airfoil design,[5] and likewise, the second sample data set typifies PIV data restoration[57,58] and basis extraction from incomplete data.[19,34] Note that the calculated missing data percentages are misleading as they do not properly represent actual missing rates with respect to a spatial domain; they are simply a ratio of missing data points with respect to the total number of data points.[84]



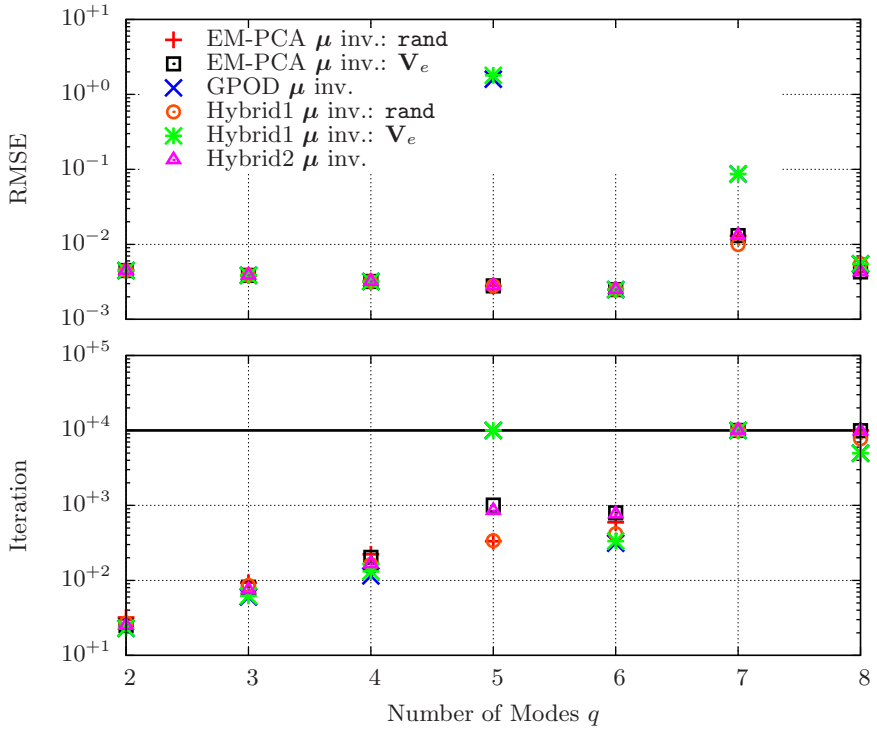
(a) μ invariant methods



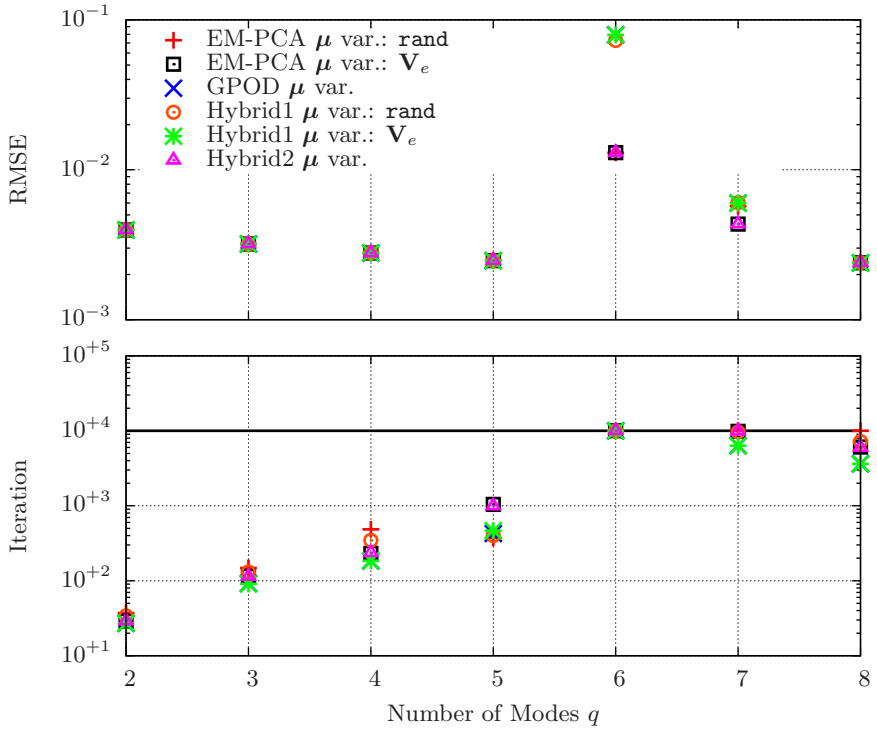
(b) μ variant methods

Figure 18: The lowest RMSEs and iteration numbers: the first sample data set whose 29.9507% of data missing only at the 57$^{\text{th}}$ Snapshot[a]

[a] A marker crossing 10^4 iterations indicates that an implementation has been aborted by the maximum iteration threshold.



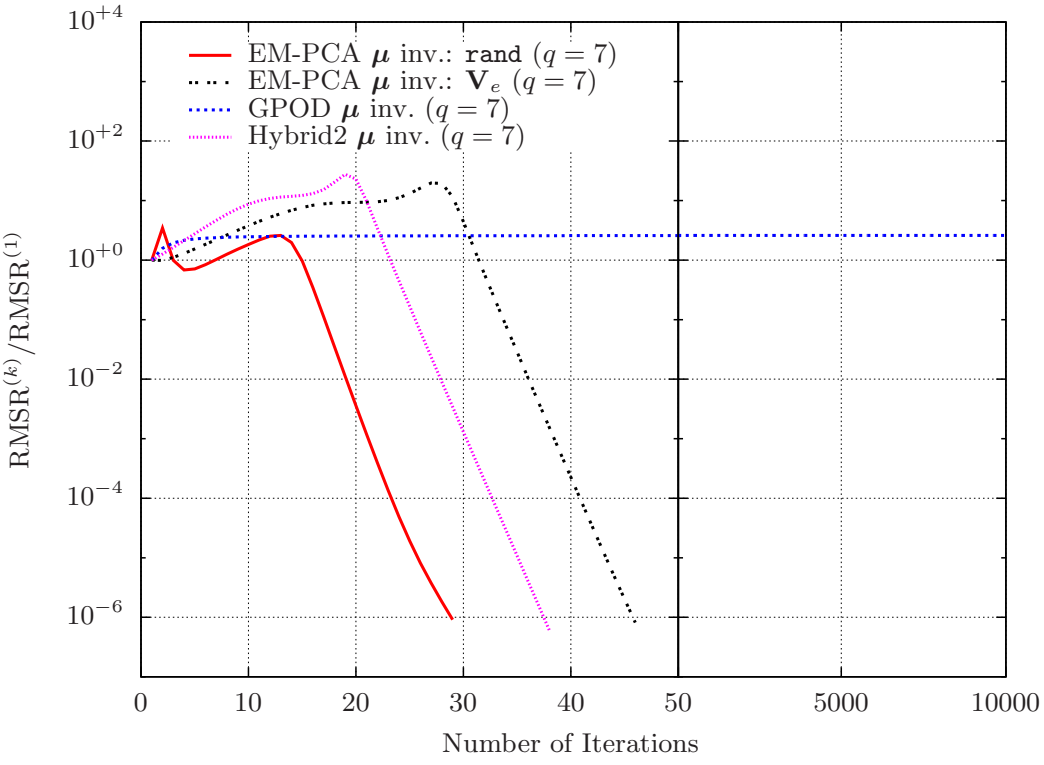
(a) $\boldsymbol{\mu}$ invariant methods



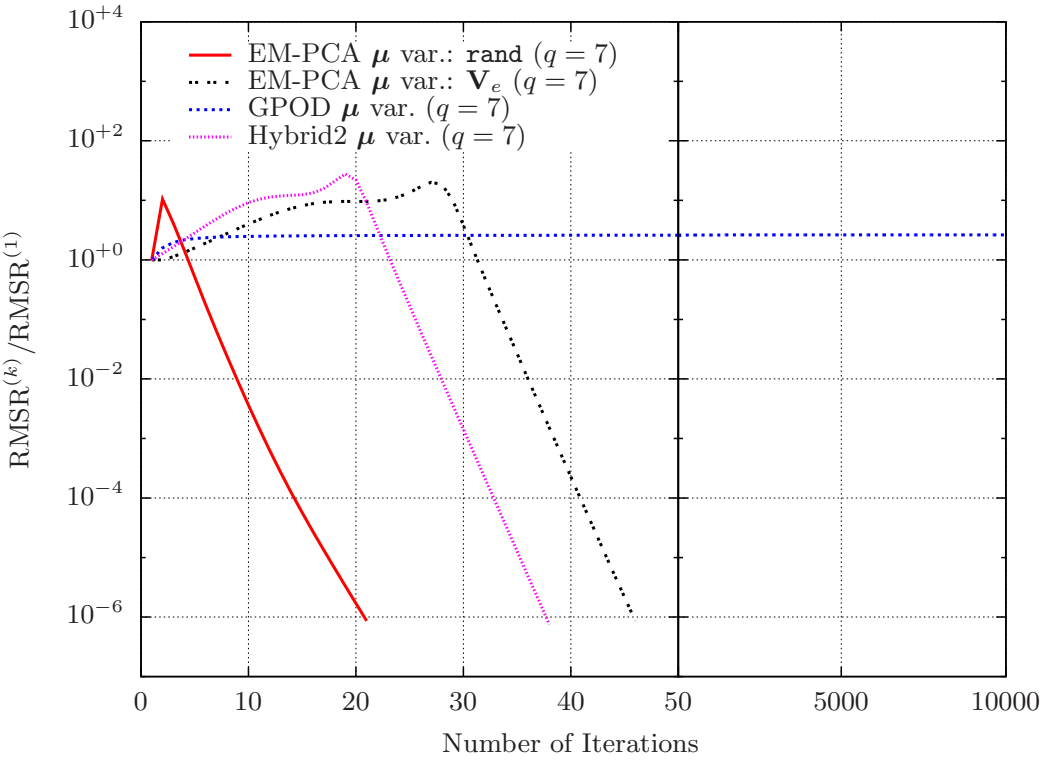
(b) $\boldsymbol{\mu}$ variant methods

Figure 19: The lowest RMSEs and iteration numbers: the second sample data set whose 29.9746% of data missing across the entire snapshot ensemble[a]

[a] A marker crossing 10^4 iterations indicates that an implementation has been aborted by the maximum iteration threshold.

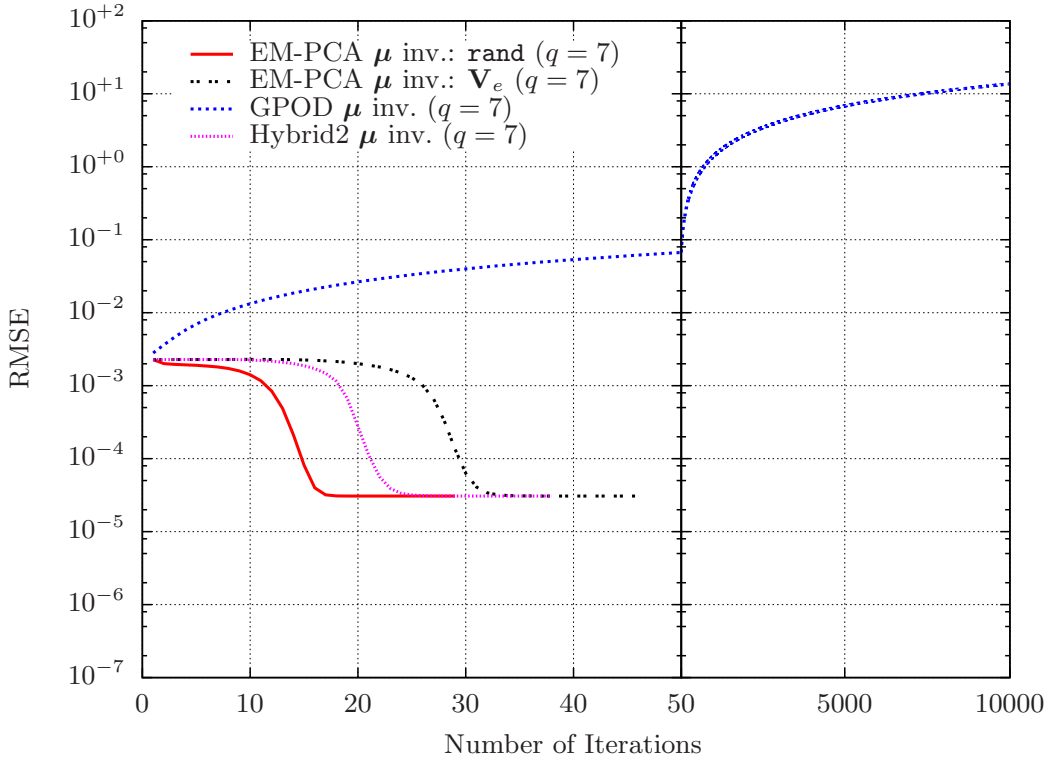


(a) $\boldsymbol{\mu}$ invariant methods

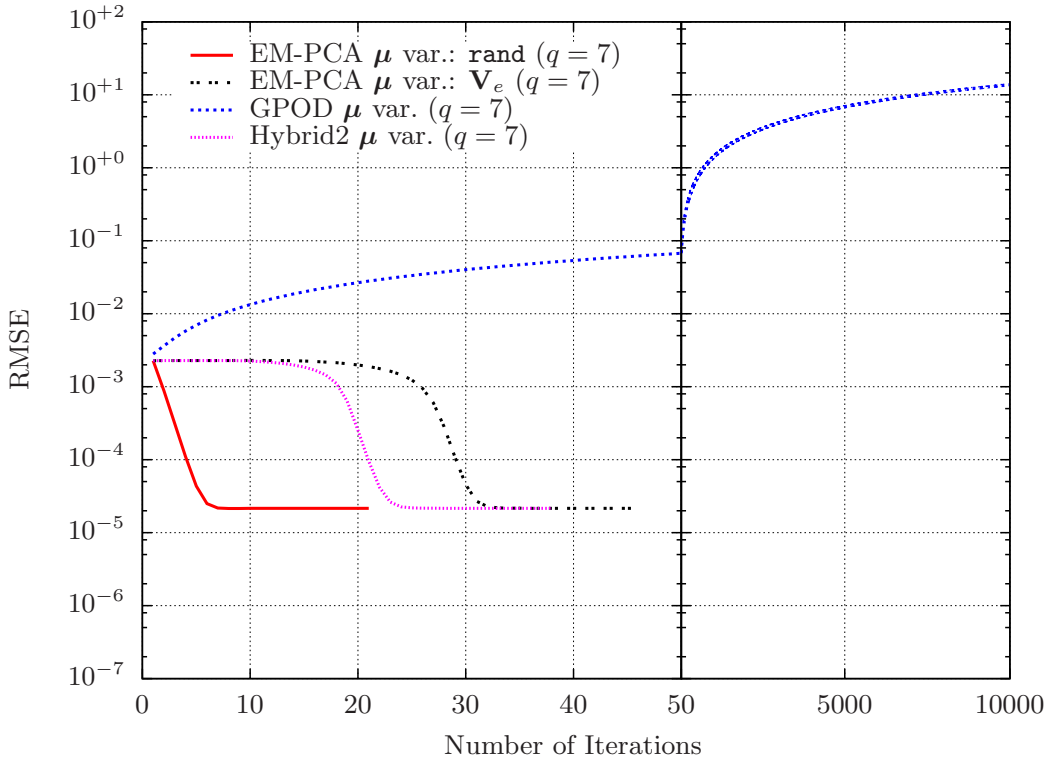


(b) $\boldsymbol{\mu}$ variant methods

Figure 20: The convergence histories of the first sample data set whose 29.9507% of data missing only at the 57$^{\text{th}}$ snapshot ($q = 7$)



(a) $\boldsymbol{\mu}$ invariant methods



(b) $\boldsymbol{\mu}$ variant methods

Figure 21: The RMSE histories of the first sample data set whose 29.9507% of data missing only at the 57^{th} snapshot ($q = 7$)

4.3.4 Selection of the Optimal Number of Modes

Because the proper number of modes q is of paramount importance to optimal reconstruction results, Lee and Mavris[30] examined RMSEs and numbers of iterations at different q values as shown in Figures 18 and 19. For instance, Figure 18 delineates RMSEs as q increases from 2 to 10 with the first sample data set, which has missing data only at the 57^{th} snapshot. As shown in Figure 18, gappy POD implementation takes fewer total numbers of iterations than the other implementations to reach the same RMSEs across all q values even though it is unstable at $q = 7$ from being terminated by the preset total iteration threshold 10^4. In particular, Figures 20 and 21 delineate the histories of normalized RMSRs and RMSEs, respectively, showing the unstable behavior of gappy POD at $q = 7$. The other implementations, using a non-orthogonal basis, take more iterations, and their numbers of total iterations start to escalate after $q = 6$ despite their stable convergence throughout all q values. Note that Hybrid 1 is left out of the numerical experiments with the first sample data set because of its numerical instability in evaluating a projection operation. Since RMSEs settle down after $q = 8$ for both "$\boldsymbol{\mu}$ inv." and "$\boldsymbol{\mu}$ var." implementations, $q = 8$ is set for further analyses with the first sample data set.

Similarly, Figure 19 depicts RMSEs for q growing from 2 to 8 with the second sample data set, which contains missing data across the entire snapshot ensemble. In Figure 19, total numbers of iterations increase gradually with q, regardless of the implementations even though RMSEs decrease steadily until $q = 6$ and $q = 5$ for "$\boldsymbol{\mu}$ inv." methods in Figure 19(a) and "$\boldsymbol{\mu}$ var." methods in Figure 19(b), respectively. When implementations are terminated by reaching the maximum number of iterations at certain high q values, they exhibit unstable convergence behavior similar to that of gappy POD implementations, depicted in Figures 20 and 21. Analogous to the previous case with the first sample data in Figure 18, gappy POD and Hybrid 1 with $\mathbf{V}_e$, both of which initialize their basis with $\mathbf{V}_e$, exhibit poor convergence behavior at $q = 5$ when they hold a sample mean in iterations. Thus, these observations suggest that $\mathbf{V}_e$ cannot reliably enhance convergence behavior as an informed basis initialization. Based on the investigation results in Figure 19, the optimal q values for the "$\boldsymbol{\mu}$ inv." and "$\boldsymbol{\mu}$ var." implementations are set to $q = 6$ and $q = 5$, respectively, for

further analyses with the second sample data set.

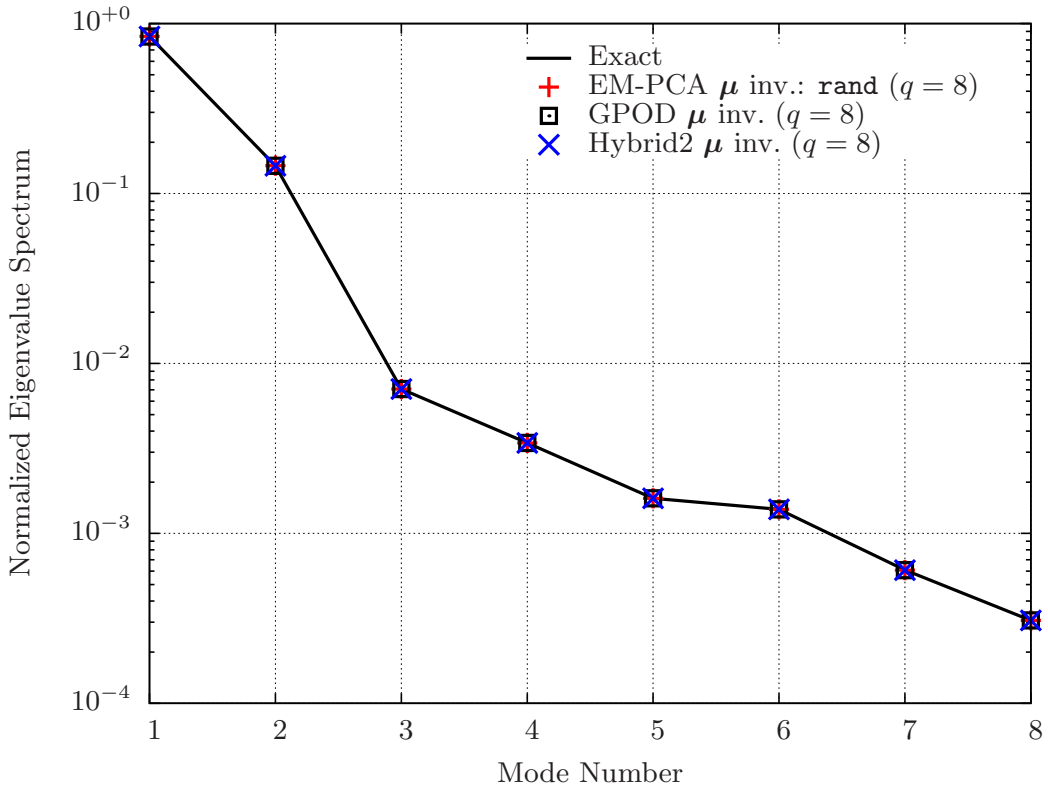
4.3.5 Eigenspectrum Validation

After the optimal q values are determined based on the lowest RMSEs in Section 4.3.4, all implementations are validated with the eigenvalues of the first and the second sample data sets. As an illustration, both Figures 22 and 23 delineate the estimated eigenvalues extracted from the two restored data sets by the implemented algorithms along with the true eigenvalues obtained from the intact data by the snapshot POD. As shown in Figure 22, in the case of the first sample data set, the estimated eigenvalues are identical to the true values regardless of the "$\boldsymbol{\mu}$ inv." and "$\boldsymbol{\mu}$ var." implementations. By contrast, in the case of the second sample data set, Figure 23(b) shows that the estimated eigenvalues by the "$\boldsymbol{\mu}$ var." implementations closely follow their corresponding exact values, as do the "$\boldsymbol{\mu}$ inv." implementations, shown in Figure 23(a), with the exception of noticeable discrepancies for the sixth eigenvalue. In particular, "GPOD $\boldsymbol{\mu}$ inv." and "Hybrid2 $\boldsymbol{\mu}$ inv.," both of which employ $\mathbf{V}_q$, compute the sixth eigenvalue more accurately than the other implementations using $\mathbf{W}$; however, since the relative contribution of the sixth mode is minuscule, no considerable differences are observed in the reconstruction results.

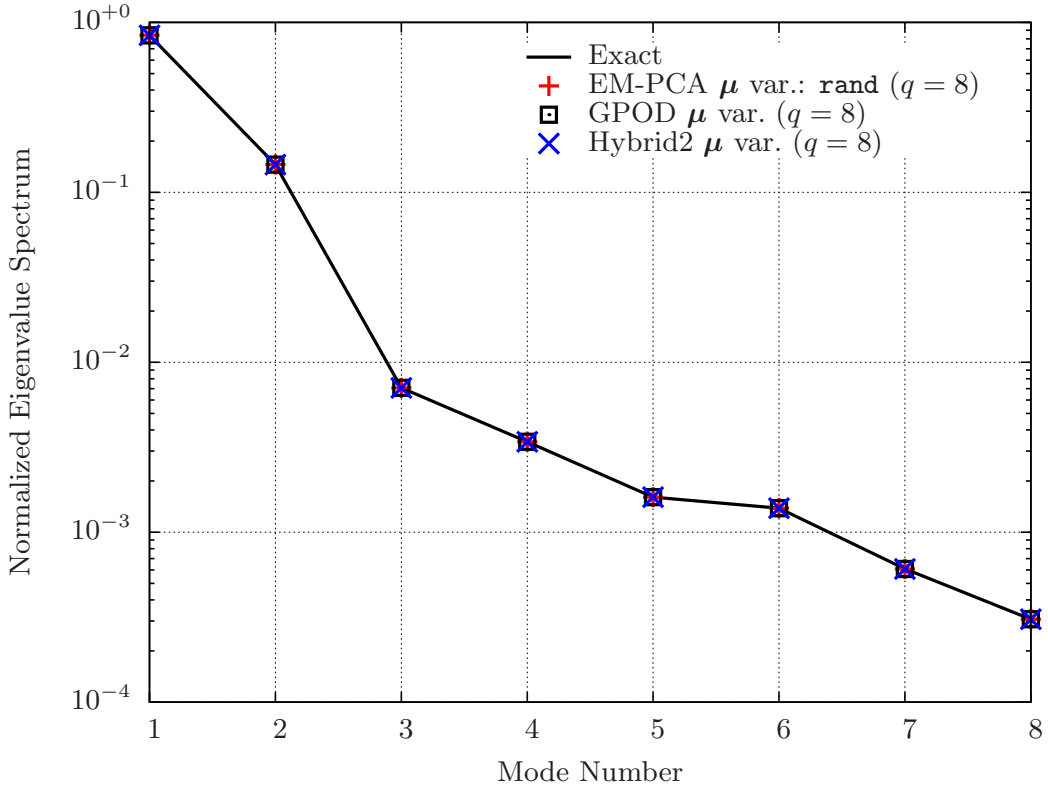
4.3.6 Quantitative Illustration of Different Basis and Norm Effects

4.3.6.1 Test Case I: Missing Data only at a Single Snapshot

Figures 24 and 25 delineate the RMSE histories of the C_p data and the basis $\mathbf{V}_q$, respectively, as the implementations reconstruct the first sample data set. Both Figures 24 and 25 list the RMSE histories obtained with two different basis initializations, namely $\mathbf{W}^{(0)} = \mathbf{V}_e$ and $\mathbf{W}^{(0)} =$ `rand` on the top and the bottom of their sub-figures, and also organize the RMSE histories according to the "$\boldsymbol{\mu}$ inv." and "$\boldsymbol{\mu}$ var." implementations. With the RMSE histories in Figures 24 and 25, the norm difference can be quantified by the comparison of the gappy POD and Hybrid 2 implementations, which share the same $\mathbf{V}_q$ basis, and similarly, the basis difference can be captured by the comparison of the EM-PCA and Hybrid 2 implementations, which share the same L^2 norm. Since RMSE differentials between the

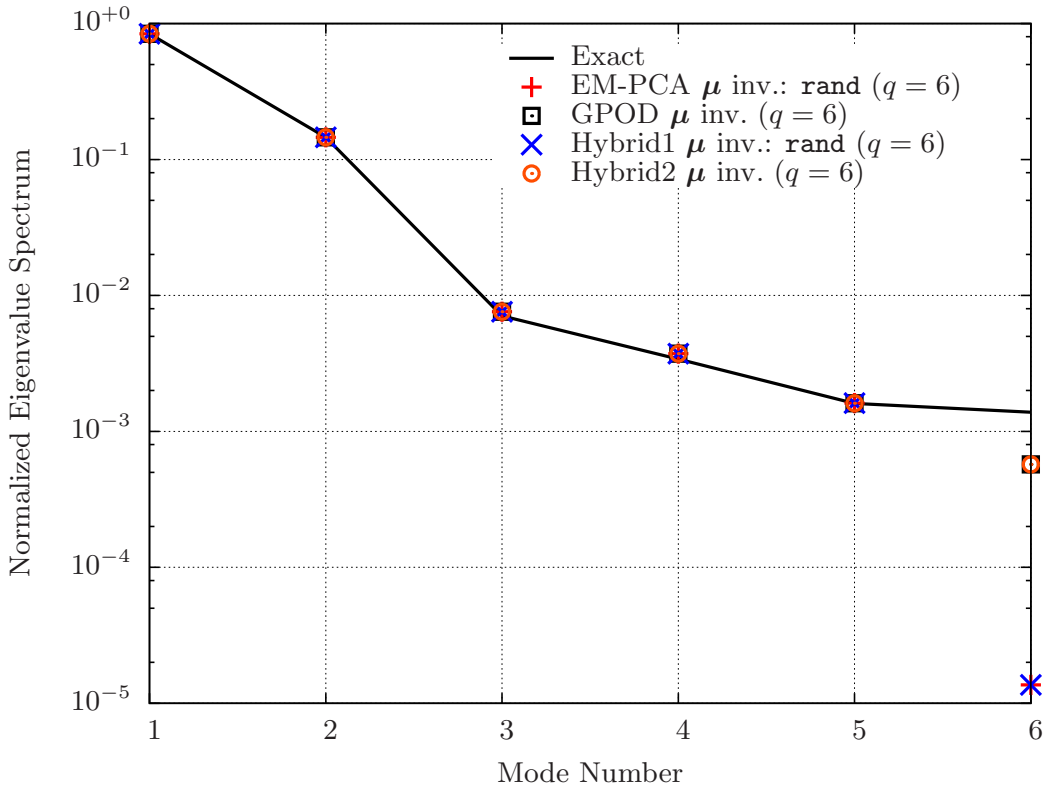


(a) $\boldsymbol{\mu}$ invariant methods

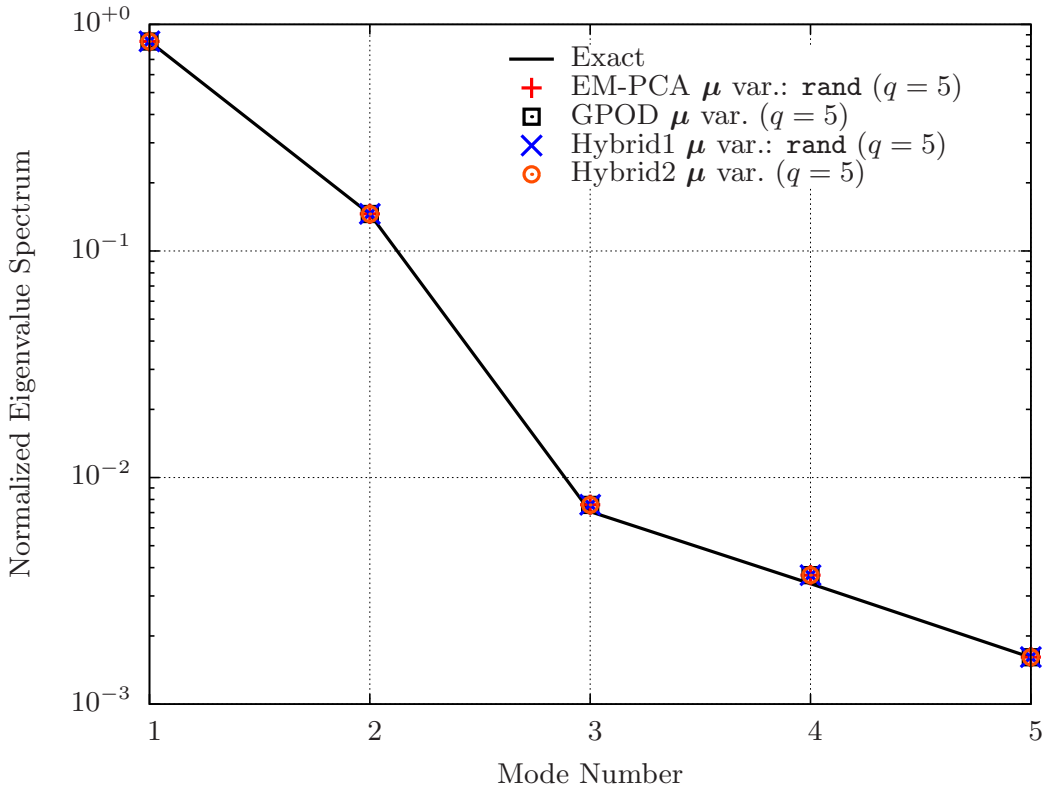


(b) $\boldsymbol{\mu}$ variant methods

Figure 22: The restored eigenvalue spectrum of the C_p data: the first sample data set whose 29.9507% of data missing only at the 57^{th} snapshot

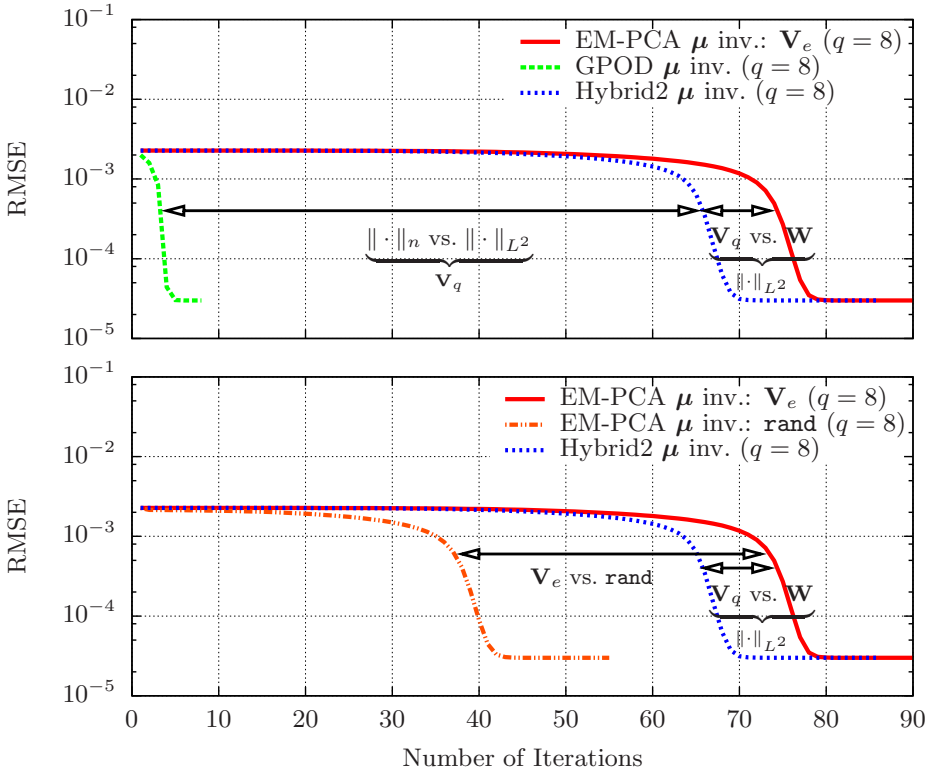


(a) $\boldsymbol{\mu}$ invariant methods

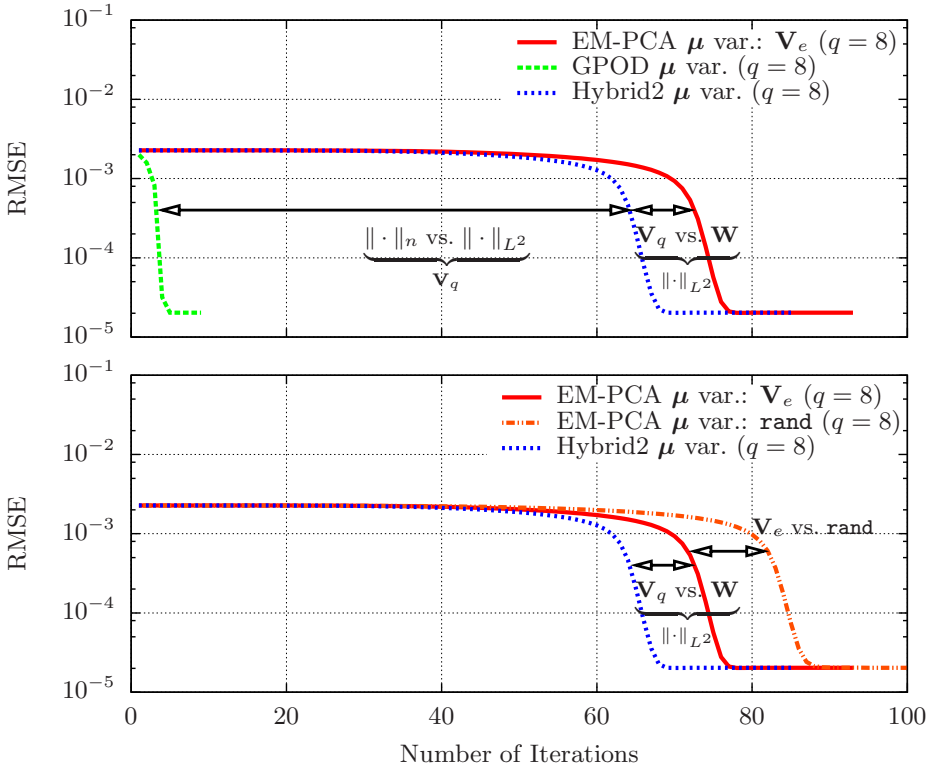


(b) $\boldsymbol{\mu}$ variant methods

Figure 23: The restored eigenvalue spectrum of the C_p data: the second sample data set whose 29.9746% of data missing across the entire snapshot ensemble

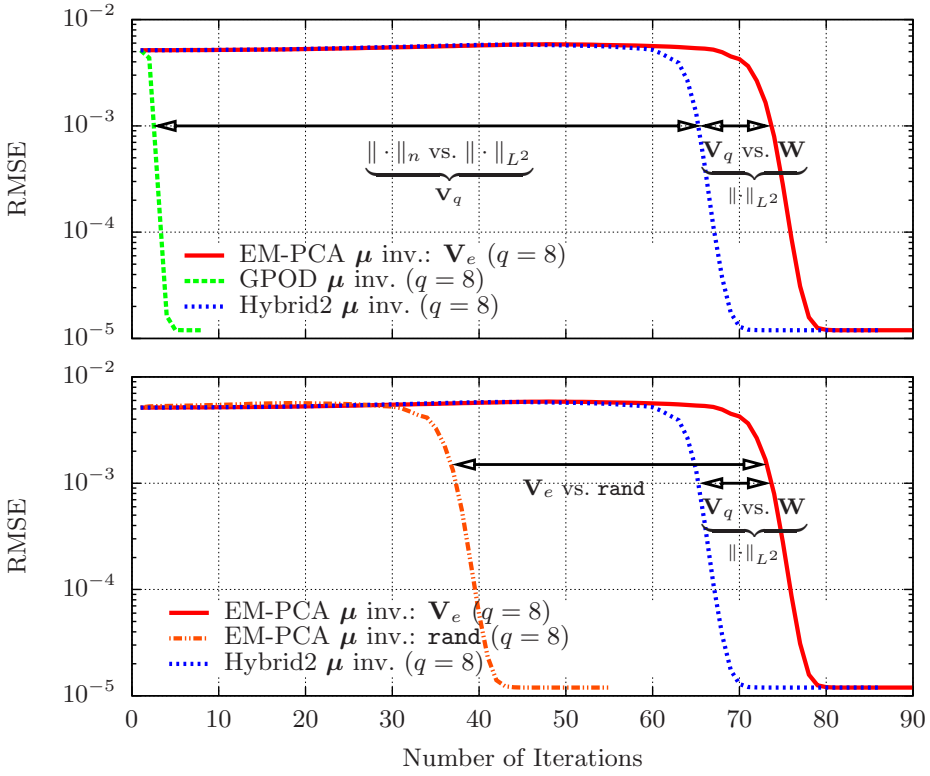


(a) $\boldsymbol{\mu}$ invariant methods

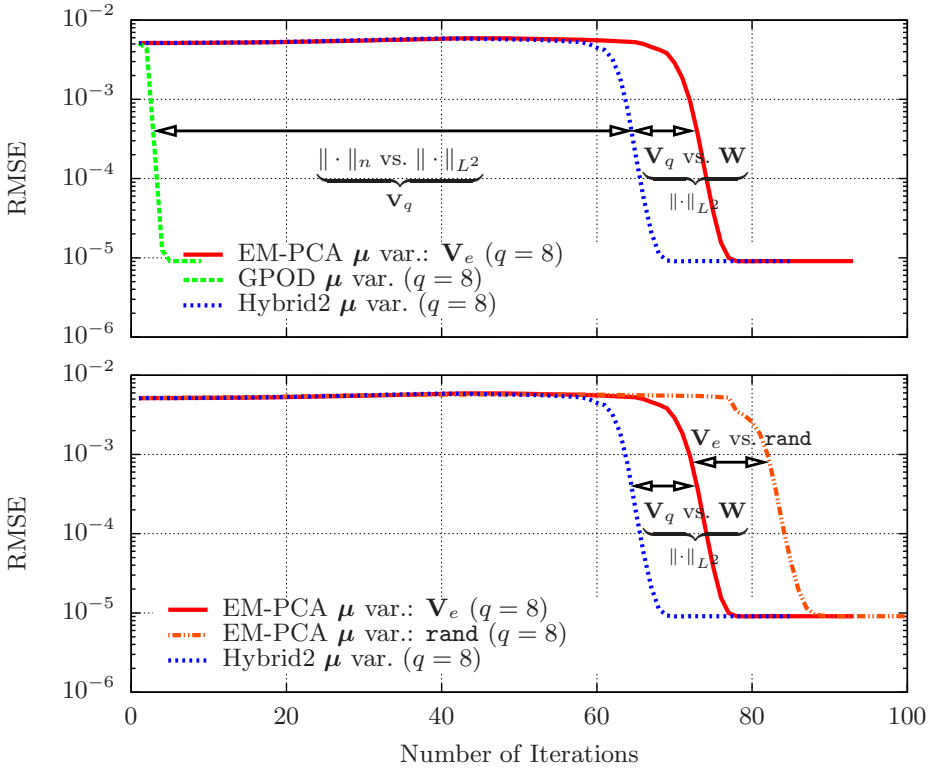


(b) $\boldsymbol{\mu}$ variant methods

Figure 24: The RMSE histories of the c_p data: the first sample data set whose 29.9507% of data missing only at the 57^{th} snapshot



(a) $\boldsymbol{\mu}$ invariant methods



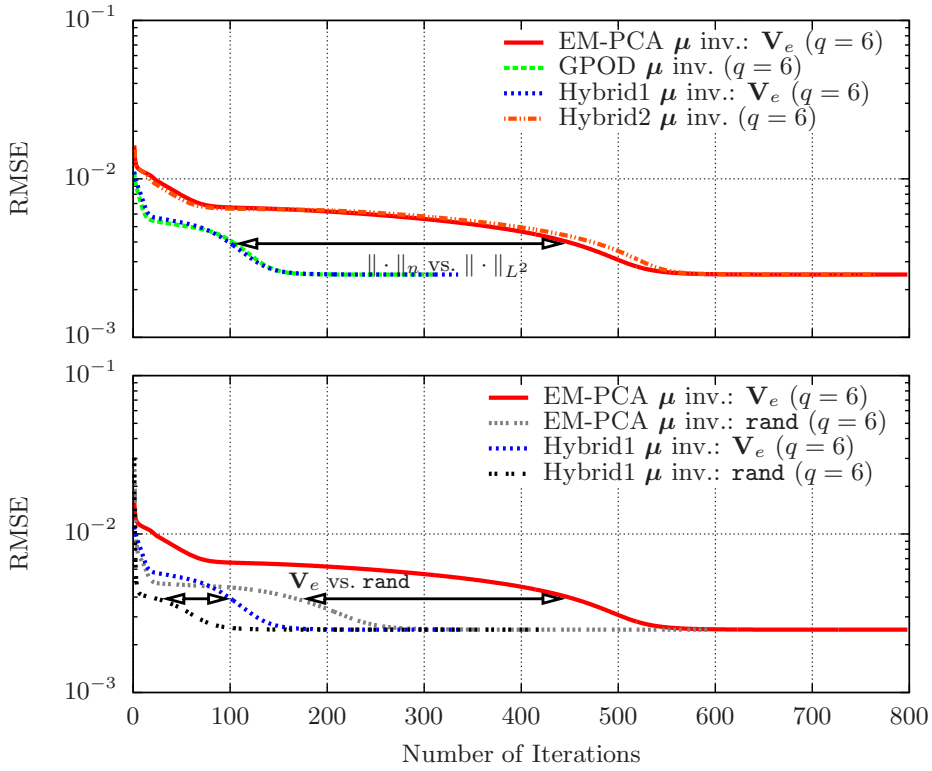
(b) $\boldsymbol{\mu}$ variant methods

Figure 25: The RMSE histories of $\mathbf{V}_q$: the first sample data set whose 29.9507% of data missing only at the 57^{th} snapshot

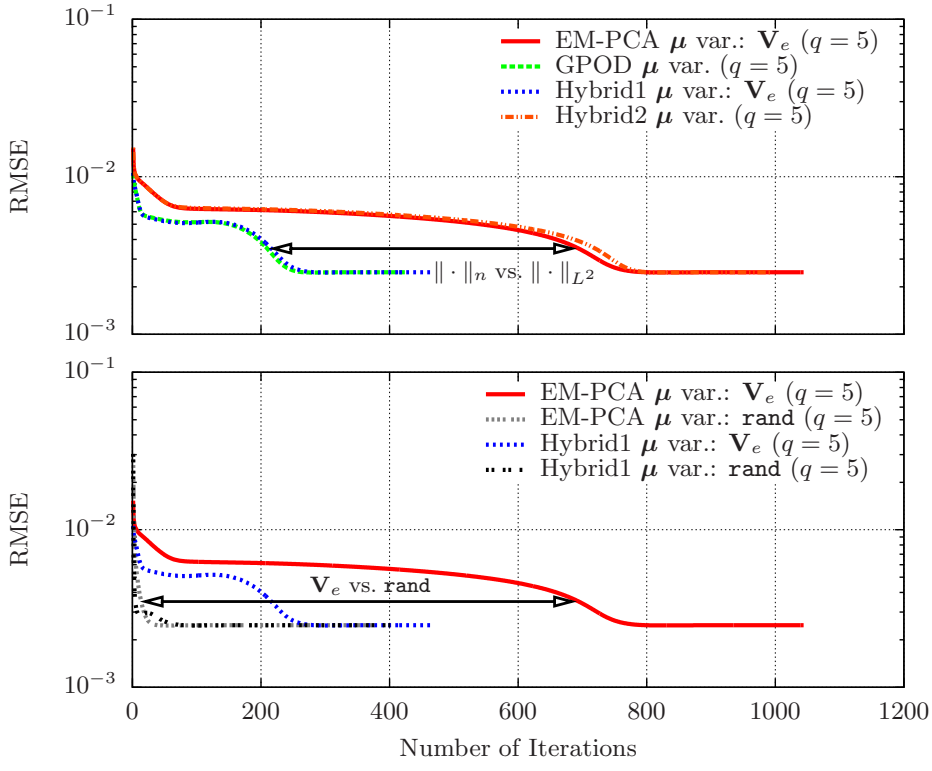
gappy POD and Hybrid 2 implementations are much larger than those between the EM-PCA and Hybrid 2 implementations, the norm effect is more conspicuous than the basis effect for restoring missing data. Because of the gappy norm, unlike Hybrid 2 using the L^2 norm, gappy POD can quickly restore missing data, resulting in a $\widetilde{\mathbf{V}}_q$ much closer to the true $\mathbf{V}_q$, as shown in Figure 25. In addition to the unbiased comparison results, on the bottom of sub-figures in Figures 24 and 25, the RMSE histories with randomly initialized $\mathbf{W}$ indicate that a random initialization can effectively reduce RMSEs in the case of the "$\boldsymbol{\mu}$ inv." implementations, shown in Figures 24(a) and 25(a).

4.3.6.2 Test Case II: Missing Data across all the Snapshots

Similar to the previous RMSE histories with the first sample data set, Figures 26 and 27 show the RMSE histories of the C_p data and the basis $\mathbf{V}_q$ with the second sample data set. In Figures 26 and 27, the RMSE histories with the $\mathbf{W}^{(0)} = \mathbf{V}_e$ initialization facilitate the identification of the different basis and norm effects on missing data estimation. First, the different basis effect can be assessed by either the comparison of the gappy POD and Hybrid 1 implementations, which share the same gappy norm, or the comparison of the EM-PCA and Hybrid 2 implementations, which share the same L^2 norm. Likewise, the different norm effect can be evaluated by either the comparison of the gappy POD and Hybrid 2 implementations, which share the same $\widetilde{\mathbf{V}}_q$, or the comparison of the EM-PCA and Hybrid 1 implementations, which share the same $\widetilde{\mathbf{W}}$. These systematic, unbiased comparative studies reveal that the norm difference again plays a dominant role in missing data estimation, just as it does for the first sample data set. In contrast to the results of the first sample data set, the basis difference barely produces differentials in the RMSE histories with the second sample data set, unlike the case of the first sample data. Overall, these quantitative analyses on the basis and norm effects with the two sample data sets reveal that the gappy POD and Hybrid 1 implementations are better at estimating missing data than the other implementations mainly because of their gappy norm. In addition, at the bottom of sub-figures in Figures 26 and 27, the RMSE histories with randomly initialized $\mathbf{W}$ show that a random initialization is more conducive to reducing RMSEs than the informed

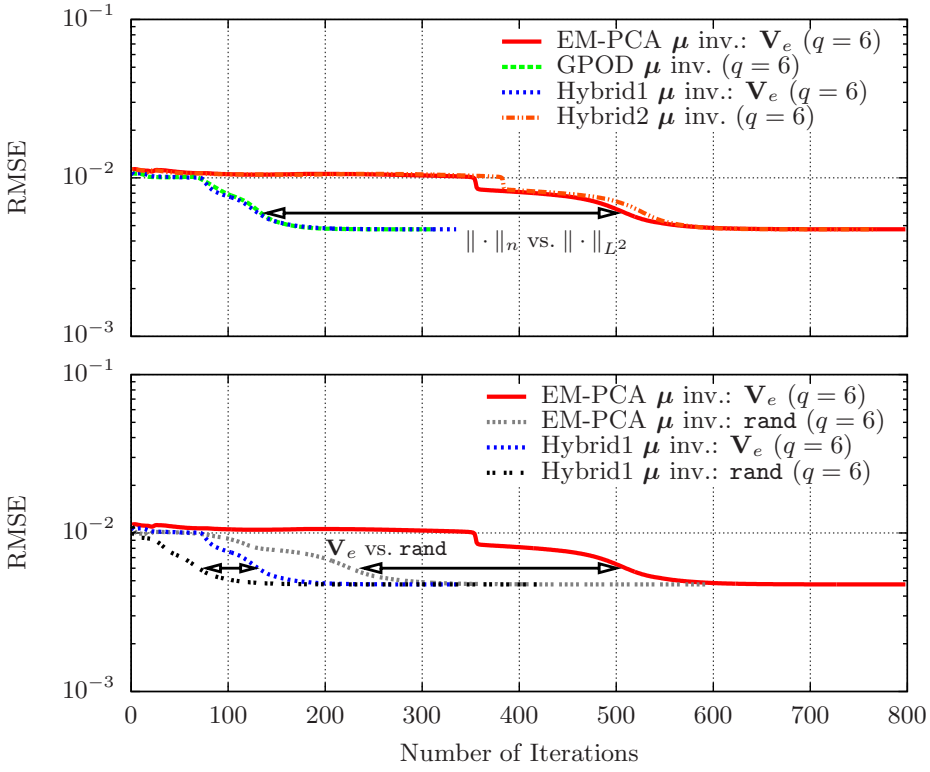


(a) $\boldsymbol{\mu}$ invariant methods

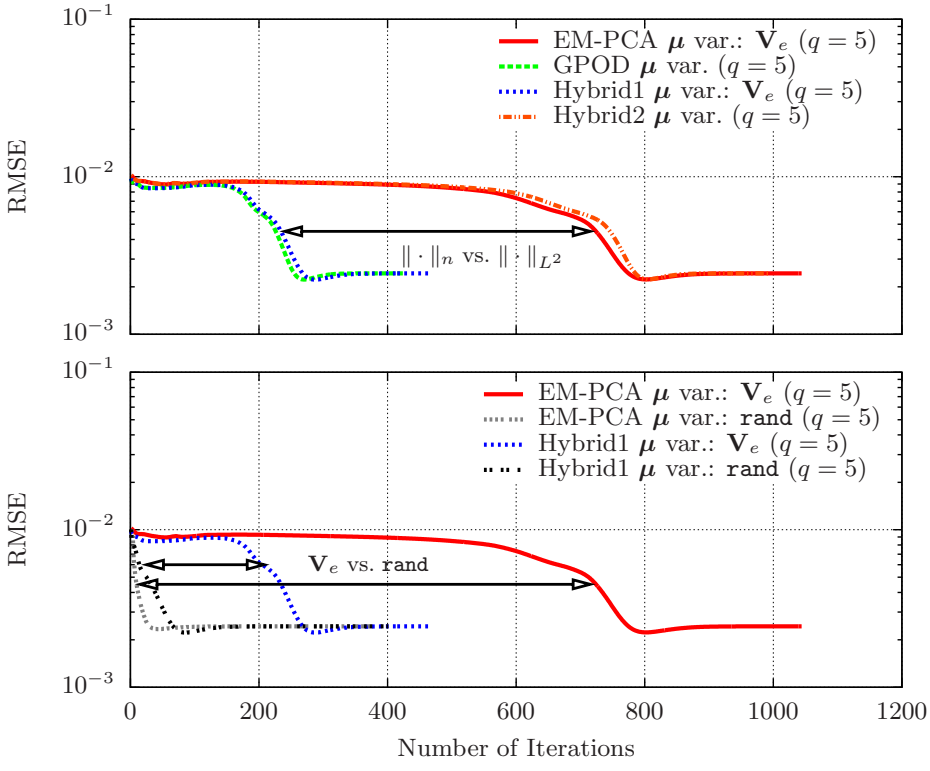


(b) $\boldsymbol{\mu}$ variant methods

Figure 26: The RMSE histories of the C_p Data: the second sample data set whose 29.9746% of data missing across the entire snapshot ensemble



(a) $\boldsymbol{\mu}$ invariant methods



(b) $\boldsymbol{\mu}$ variant methods

Figure 27: The RMSE histories of $\mathbf{V}_q$: the second sample data set whose 29.9746% of data missing across the entire snapshot ensemble

initialization of $\mathbf{W}$ with $\mathbf{V}_e$. Moreover, the comparison of the RMSE histories in Figures 26 and 27 show that the different $\mathbf{W}$ initializations, $\mathbf{V}_e$ or `rand`, which may be more beneficial for missing data estimation than the different basis types, $\widetilde{\mathbf{W}}$ or $\widetilde{\mathbf{V}}_q$.

4.3.6.3 Summary of the Quantitative Investigation

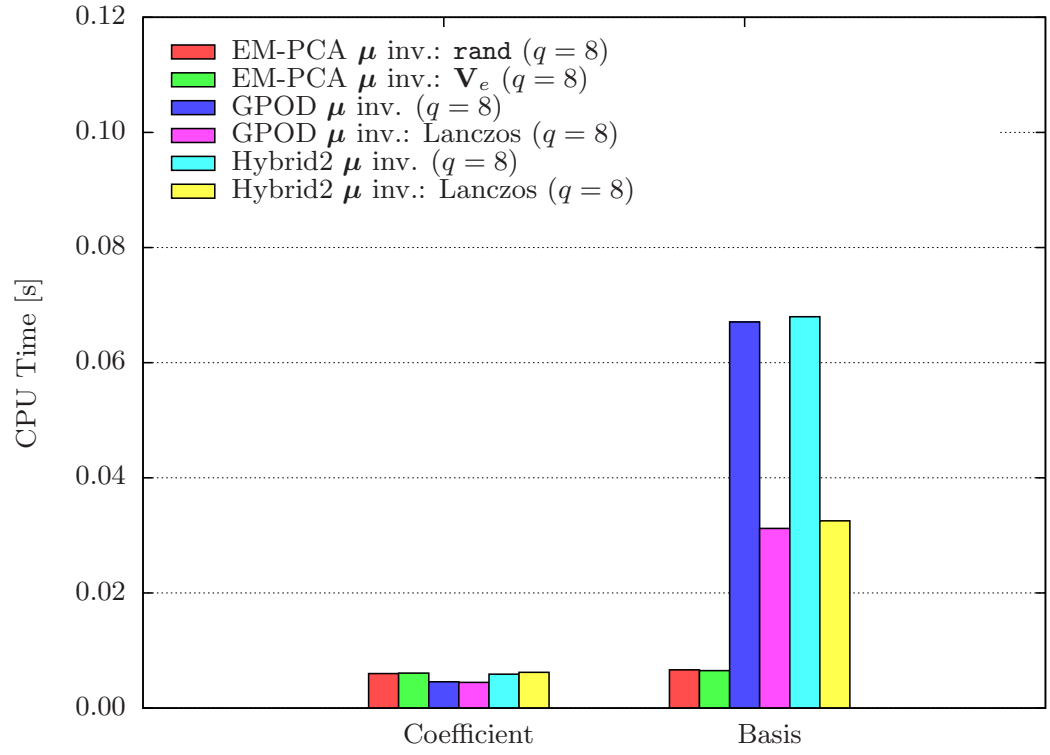
In short, the previous observations with the two test cases clearly identify that norm selection more strongly affects missing data estimation than basis selection. Of the two different bases, the RMSE histories illustrate that the benefit of $\widetilde{\mathbf{V}}_q$ over $\widetilde{\mathbf{W}}$ is inconsistent, varying with the data-missing characteristics of the given incomplete data sets. For instance, for the first sample data set, $\widetilde{\mathbf{V}}_q$ is more effective than $\widetilde{\mathbf{W}}$ to some degree, but for the second sample data set, it is almost equally as effective as $\widetilde{\mathbf{W}}$. Next, of the two different norms, the RMSE histories evidently substantiate that the gappy norm contributes to RMSE drops more significantly than the L^2 norm does for both sample data sets. Last, a random initialization for $\mathbf{W}$ could have more influence than the use of $\widetilde{\mathbf{V}}_q$ if the given data set has evenly scattered missing data, like the second sample data set. Note that the poor convergence performance observed in the RMSEs with $\mathbf{V}_e$ initialization is largely caused by scattered missing data that prevents $\widetilde{\mathbf{Y}}^{(0)}$, which $\mathbf{V}_e$ relies on, from approaching close to the intact $\mathbf{Y}$.

4.4 Computational Efficiency Comparison

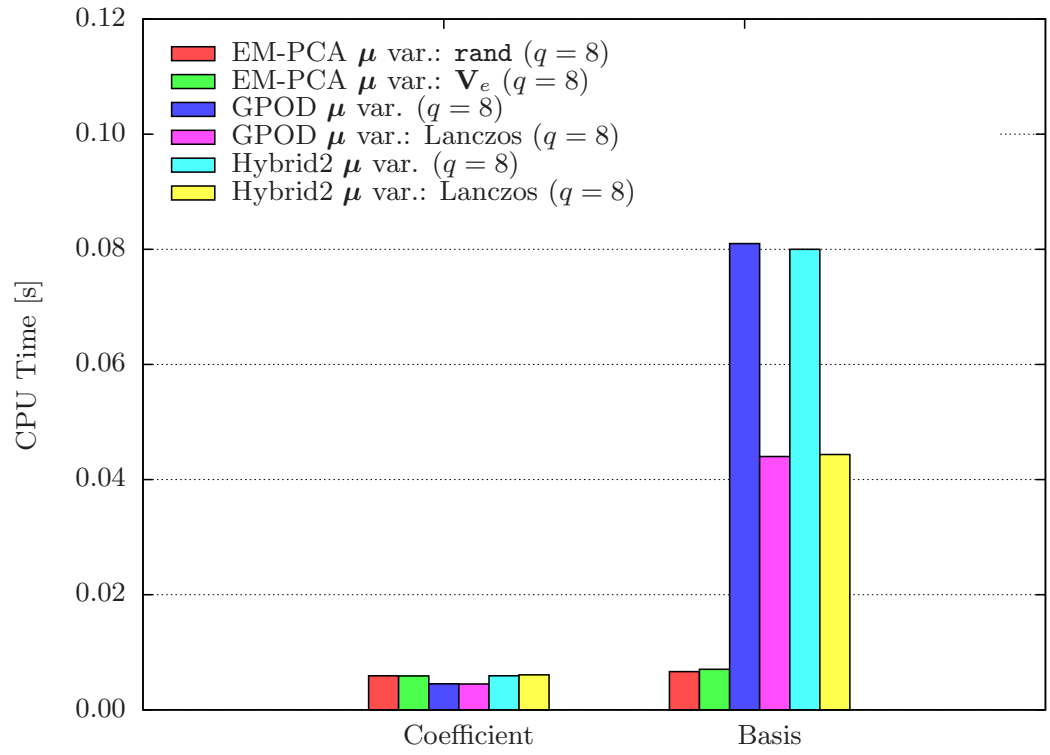
Before the investigation of the performance of the implemented algorithms, this section illustrates the numerical cost for each basis and coefficient evaluation in conjunction with their formulations described earlier in Tables 2 and 3. Afterwards, it discusses the overall computational performance of the implementations, analyzed in terms of their basis and coefficient evaluations.

4.4.1 Computational Cost for a Basis and Coefficient Evaluation

The basis and coefficient formulations in Table 2 suggest that each basis and coefficient evaluation of the EM-PCA takes less effort than that of gappy POD. In order to verify the anticipated evaluation costs in Section 4.1.2.1, this research measured the computational

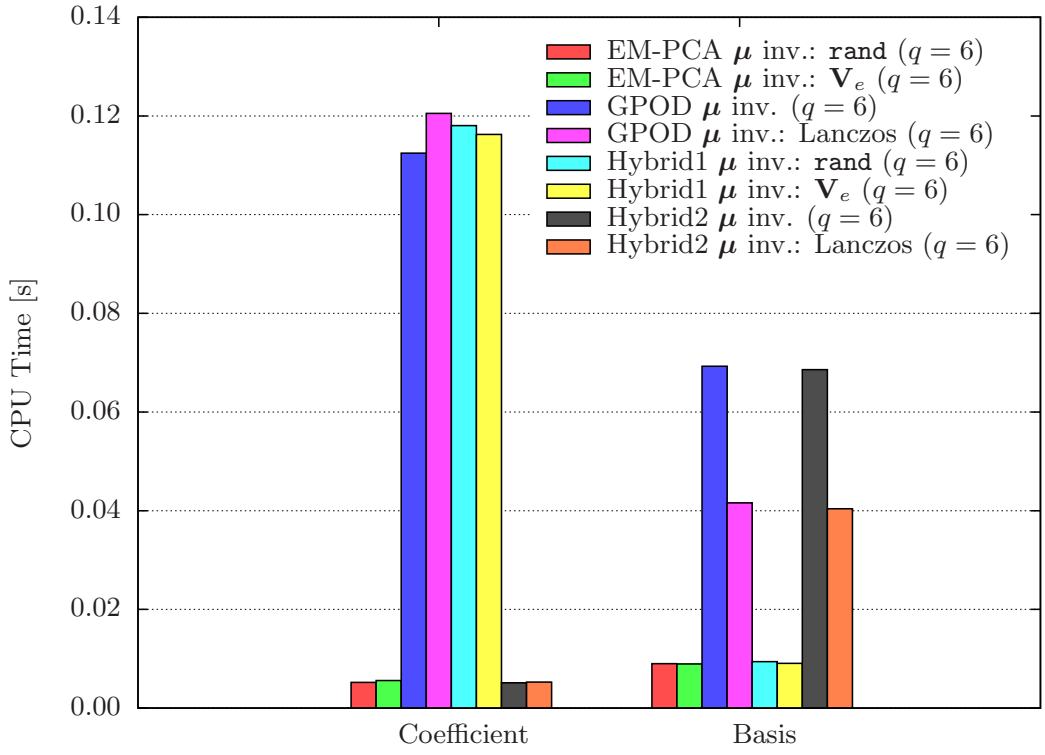


(a) $\boldsymbol{\mu}$ invariant methods

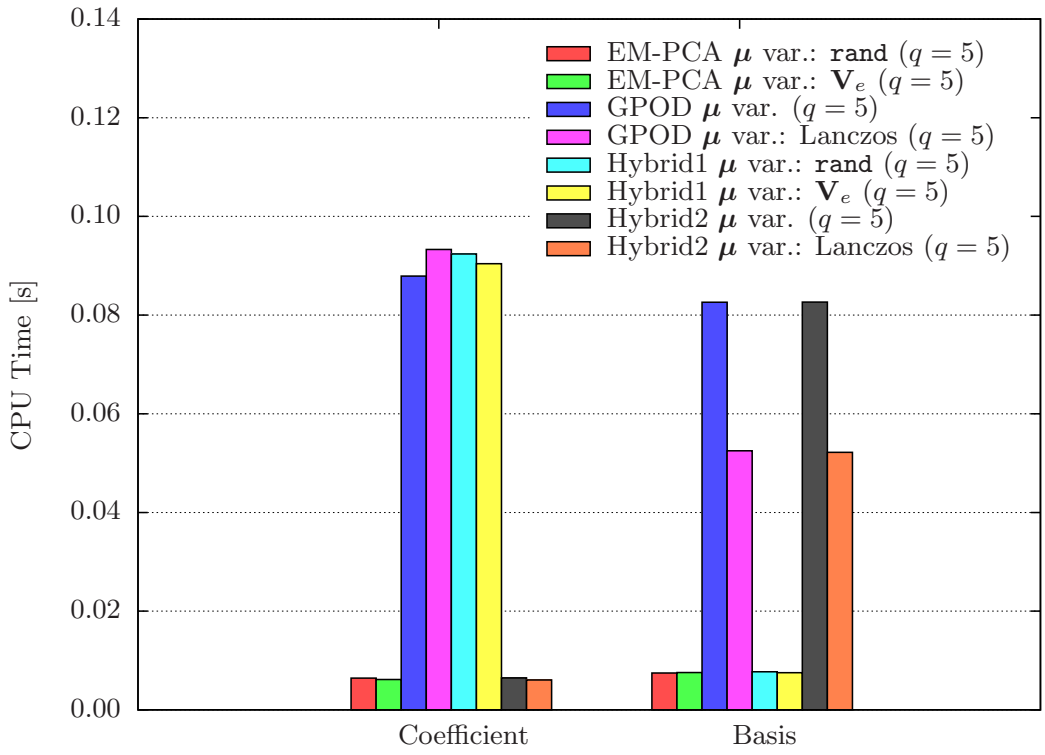


(b) $\boldsymbol{\mu}$ variant methods

Figure 28: Computational time for a single basis and coefficient evaluation: the first sample data set whose 29.9507% of data missing only at the 57^{th} snapshot



(a) $\boldsymbol{\mu}$ invariant methods



(b) $\boldsymbol{\mu}$ variant methods

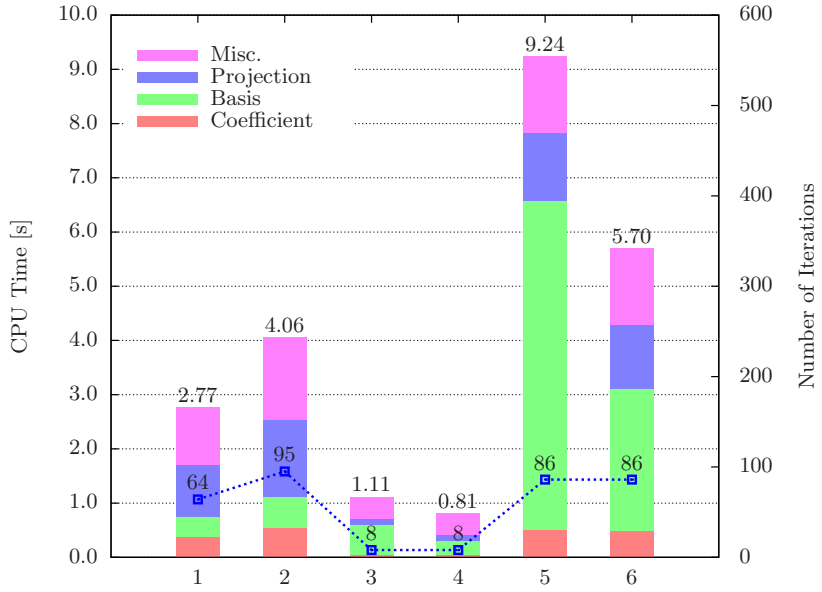
Figure 29: Computational time for a single basis and coefficient evaluation: the second sample data set whose 29.9746% of data missing across the entire snapshot ensemble

time spent for the single evaluation of a basis and a coefficient using the two sample data sets. First, with regard to the single evaluation of a basis, Figures 28 and 29 confirm that $\mathbf{W}$ for the EM-PCA and Hybrid 1 takes less time than $\mathbf{V}_q$ for gappy POD and Hybrid 2. Of the "$\boldsymbol{\mu}$ inv." and "$\boldsymbol{\mu}$ var." implementations, the basis evaluation of the latter in Figures 28(b) and 29(b) is slightly slower than that of the former in Figures 28(a) and 29(a) since it involves subtracting a sample mean to account for sample mean changes. Note that even the Lanczos algorithm, which expedites the evaluation of $\mathbf{V}_q$, cannot outperform that of $\mathbf{W}$. Second, with regard to the single evaluation of a coefficient, Figure 28 demonstrates no computational time differences in evaluating a coefficient with respect to the norm difference because the number of data-missing snapshots is only one. However, Figure 29 shows that the computation time entailed by $\mathbf{P}_{\text{GPOD}}$, used by the EM-PCA and Hybrid 2, is in stark contrast to that of $\mathbf{P}_{\text{EM-PCA}}$, used by gappy POD and Hybrid 1, due to multiple data-missing snapshots. Since $\mathbf{P}_{\text{GPOD}}$ is induced by the gappy norm, it necessitates repetitive evaluations for every different $\mathring{\mathbf{y}}_j$; however, $\mathbf{P}_{\text{EM-PCA}}$, resulting from the L^2 norm, does not.

4.4.2 Computational Time Breakdown with the Number of Iterations

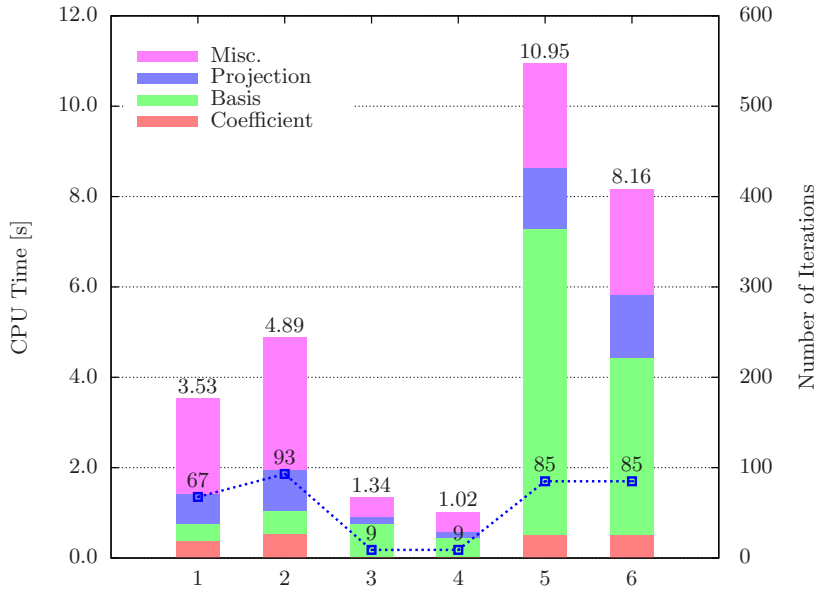
As an illustration of the overall numerical performance of all the implementations, both Figures 30 and 31 illustrate total computational time along with total numbers of iterations with the first and the second sample data sets, respectively. In addition, the total computational time is decomposed to show computational times spent at evaluating bases and coefficients for all the implementations. For those implementations randomly initializing $\mathbf{W}$, their test results were averaged over 100 runs for the minimal randomness effect. In Figure 30, the test results with the first sample data reveal that the gappy POD implementations take much fewer iterations and less computational time than the other implementations, owing to the gappy norm. Although the EM-PCA and Hybrid 2 implementations, which share the L^2 norm, require approximately the same number of iterations, the Hybrid 2 implementations take more computational time than the EM-PCA implementations because of their computationally expensive $\widetilde{\mathbf{V}}_q$ compared to $\widetilde{\mathbf{W}}$.

Next, Figure 31 with the second sample data set shows that the norm difference again



1: EM-PCA $\boldsymbol{\mu}$ inv.: `rand` ($q = 8$)
2: EM-PCA $\boldsymbol{\mu}$ inv.: $\mathbf{V}_e$ ($q = 8$)
3: GPOD $\boldsymbol{\mu}$ inv. ($q = 8$)
4: GPOD $\boldsymbol{\mu}$ inv.: Lanczos ($q = 8$)
5: Hybrid2 $\boldsymbol{\mu}$ inv. ($q = 8$)
6: Hybrid2 $\boldsymbol{\mu}$ inv.: Lanczos ($q = 8$)

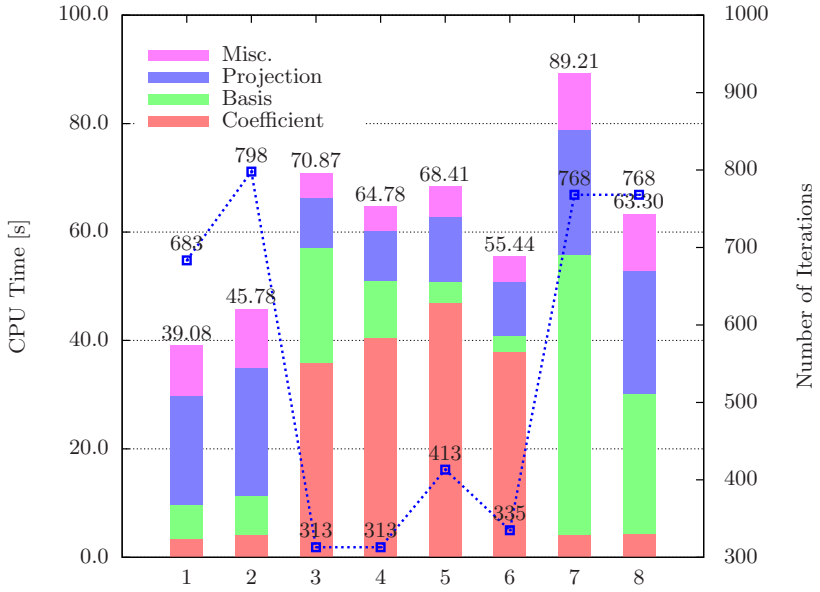
(a) $\boldsymbol{\mu}$ invariant methods



1: EM-PCA $\boldsymbol{\mu}$ var.: `rand` ($q = 8$)
2: EM-PCA $\boldsymbol{\mu}$ var.: $\mathbf{V}_e$ ($q = 8$)
3: GPOD $\boldsymbol{\mu}$ var. ($q = 8$)
4: GPOD $\boldsymbol{\mu}$ var.: Lanczos ($q = 8$)
5: Hybrid2 $\boldsymbol{\mu}$ var. ($q = 8$)
6: Hybrid2 $\boldsymbol{\mu}$ var.: Lanczos ($q = 8$)

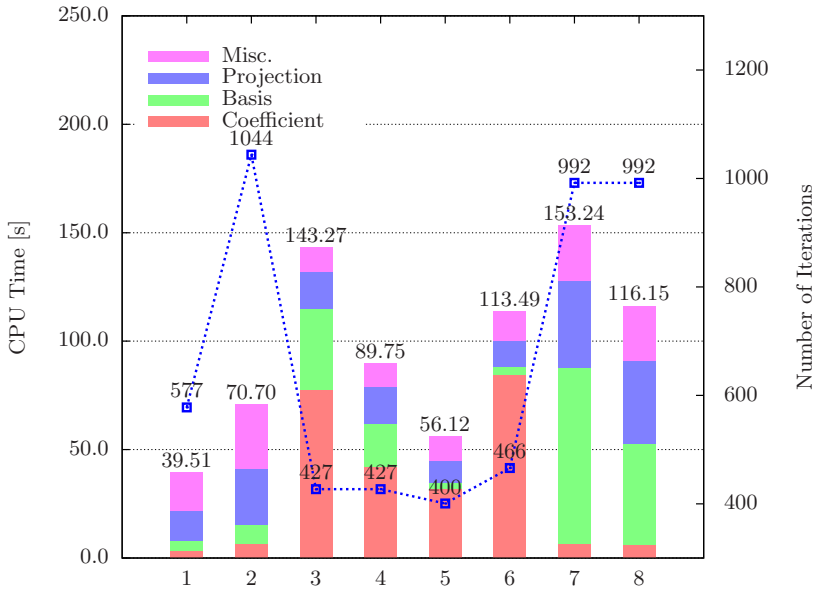
(b) $\boldsymbol{\mu}$ variant methods

Figure 30: Computational time decomposition versus iteration numbers: the first sample data set whose 29.9507% of data missing only at the 57^{th} snapshot



1: EM-PCA $\boldsymbol{\mu}$ inv.: `rand` ($q = 6$)
2: EM-PCA $\boldsymbol{\mu}$ inv.: $\mathbf{V}_e$ ($q = 6$)
3: GPOD $\boldsymbol{\mu}$ inv. ($q = 6$)
4: GPOD $\boldsymbol{\mu}$ inv.: Lanczos ($q = 6$)
5: Hybrid1 $\boldsymbol{\mu}$ inv.: `rand` ($q = 6$)
6: Hybrid1 $\boldsymbol{\mu}$ inv.: $\mathbf{V}_e$ ($q = 6$)
7: Hybrid2 $\boldsymbol{\mu}$ inv. ($q = 6$)
8: Hybrid2 $\boldsymbol{\mu}$ inv.: Lanczos ($q = 6$)

(a) $\boldsymbol{\mu}$ invariant methods



1: EM-PCA $\boldsymbol{\mu}$ var.: `rand` ($q = 5$)
2: EM-PCA $\boldsymbol{\mu}$ var.: $\mathbf{V}_e$ ($q = 5$)
3: GPOD $\boldsymbol{\mu}$ var. ($q = 5$)
4: GPOD $\boldsymbol{\mu}$ var.: Lanczos ($q = 5$)
5: Hybrid1 $\boldsymbol{\mu}$ var.: `rand` ($q = 5$)
6: Hybrid1 $\boldsymbol{\mu}$ var.: $\mathbf{V}_e$ ($q = 5$)
7: Hybrid2 $\boldsymbol{\mu}$ var. ($q = 5$)
8: Hybrid2 $\boldsymbol{\mu}$ var.: Lanczos ($q = 5$)

(b) $\boldsymbol{\mu}$ variant methods

Figure 31: Computational time decomposition versus iteration numbers: the second sample data set whose 29.9746% of data missing across the entire snapshot ensemble

results in significant differentials in the total number of iterations, more than the basis difference as seen in Figure 30. Nonetheless, unlike Figure 30, the total number of iterations is not commensurate with the resulting computational performance as shown in Figure 31. For example, due to their gappy norm, the gappy POD and Hybrid 1 implementations take much fewer iterations than the other EM-PCA and Hybrid 2 implementations. However, despite the lower iteration numbers of those implementations with the gappy norm, they take more computational time than the EM-PCA and Hybrid 2 implementations, both of which use the L^2 norm. This imbalance between computational time and iteration numbers is mainly caused by the coefficient evaluations, which reflect the norms as delineated in Figure 31. For the second sample data set, implementations with the gappy norm require a new evaluation of $\mathbf{P}_{\text{GPOD}}$ for every missing snapshot compared to the those with the L^2 norm, involving a constant $\mathbf{P}_{\text{EM-PCA}}$ to any missing snapshots. Note that an accelerated $\widetilde{\mathbf{V}}_q$ evaluation by the Lanczos algorithm cannot produce much benefit for the gappy POD and Hybrid 2 implementations; for those implementations, the basis difference has a relatively insignificant effect compared to the norm difference on their total computational time.

Overall, both numerical performance tests in Figures 30 and 31 reveal that the norm difference is a key factor that significantly affects both computational time and iteration number, compared to the basis difference. Due to the gappy norm, the gappy POD implementations are found to be the most efficient for the first sample data set, which has missing data only in a single snapshot. In contrast, because of the L^2 norm, the EM-PCA implementations are computationally faster than the gappy POD implementations for the second sample data set, which has missing data across all the snapshots. Last, a random initialization for $\mathbf{W}$ can be more computational beneficial on average than an informed initialization for $\mathbf{W}$ with $\mathbf{V}_e$ as demonstrated with the two sample data sets in Figures 30 and 31.

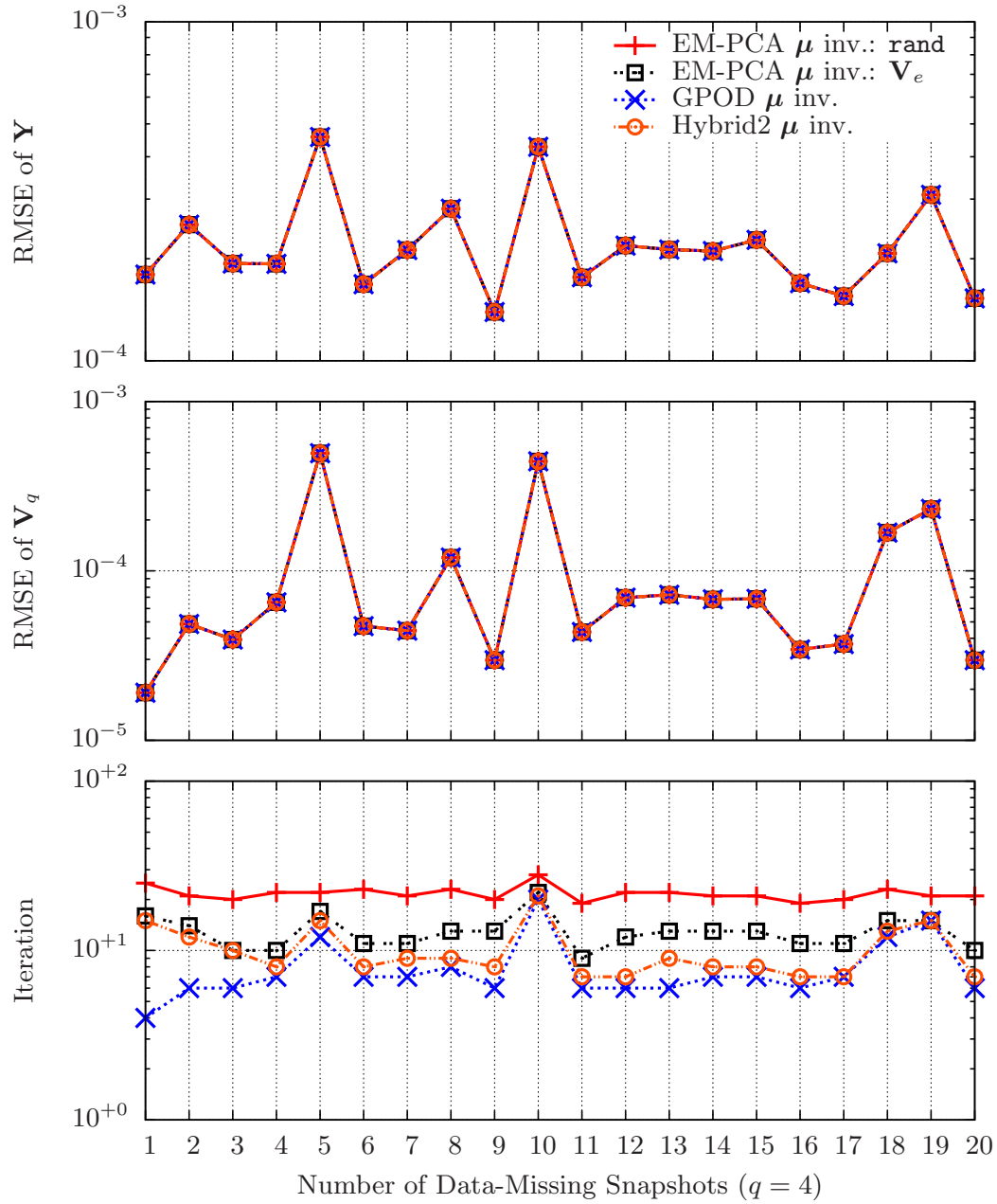
4.4.3 Performance Variations with the Increase of the Number of Data-Missing Snapshots

In the previous Section 4.4.2, the computational performance of all the implementations is demonstrated with two extreme sample data sets; one has missing data in only a single

snapshot in Figure 30, and the other has missing data in all the snapshots in Figure 31. In order to conjecture the numerical performance for other in-between cases of the two sample data sets, Lee and Mavris[30] scrutinizes the relationship between RMSEs and total numbers of iterations as the number of data-missing snapshots increases from 1 to 20. For this quantitative experiment, sample data sets are generated so that they have a fixed missing data rate of 30%, and to test the implementations, q is set to 4. As illustrated in Figure 32, in terms of iteration numbers, the EM-PCA implementation with random initialization shows fairly consistent performance behavior regardless of the sample data sets. In contrast, the other tested implementations exhibit fluctuations in the iteration numbers that are closely correlated with oscillations in RMSEs. For instance, when the numbers of data-missing snapshots are 5, 10, 18, and 19, except for the EM-PCA with random initialization, the remaining implementations struggle for convergence, resulting in large iteration numbers and low RMSEs. Such poor performance observed in particular sample data sets seems to originate from missing data present in relatively high-nonlinear snapshots with local shock phenomena. As an illustration, Figure 33 depicts two highly nonlinear snapshots that are part of the aforementioned sample data sets; for example, the 24^{th} snapshot in Figure 33(a) and the 37^{th} snapshot in Figure 33(b) are included in the sample data sets, whose numbers of data-missing snapshots are 19 and 10, respectively. After all, the EM-PCA with random initialization is more robust than the other implementations in dealing with missing data sets whose nonlinear regions are absent of data.

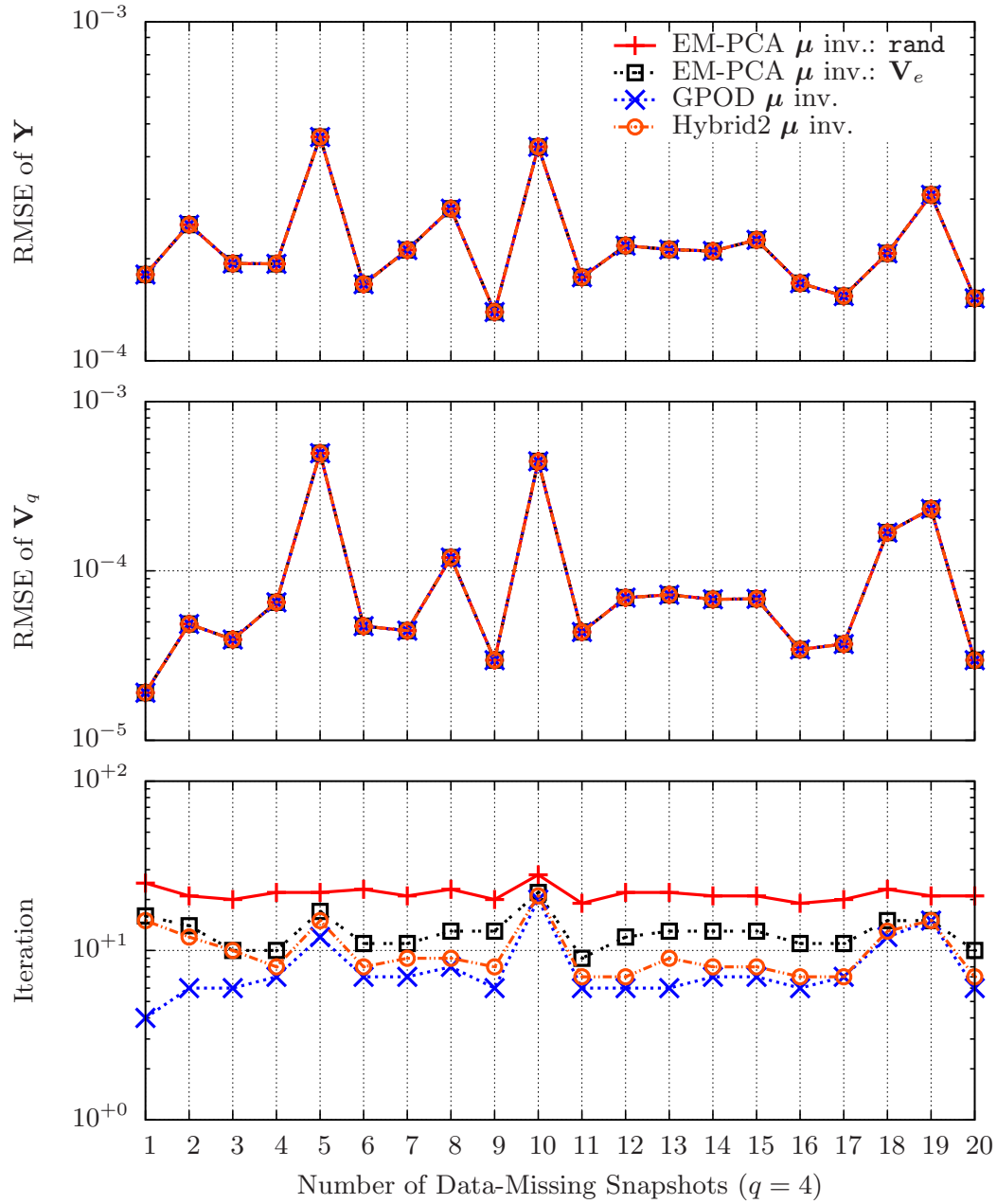
4.4.4 Computational Cost of Algorithms Expected from their Bases and Norms

The original and hybrid algorithms in tables 2 and 3, respectively, consist of dissimilar basis and coefficient evaluations directly affected by their different basis and norm selections. Regarding a basis choice from a computational aspect, the evaluation of a non-orthogonal basis $\mathbf{W}$ by matrix multiplication and inversion is much simpler than that of an orthogonal basis $\mathbf{V}_q$ by POD methods invoking SVD or EVD. For the evaluation of $\mathbf{V}_q$, the Lanczos algorithm can expedite a POD process, but its benefit is mostly limited to a small number of modes q; moreover, even at a small q, POD with the Lanczos algorithm requires more



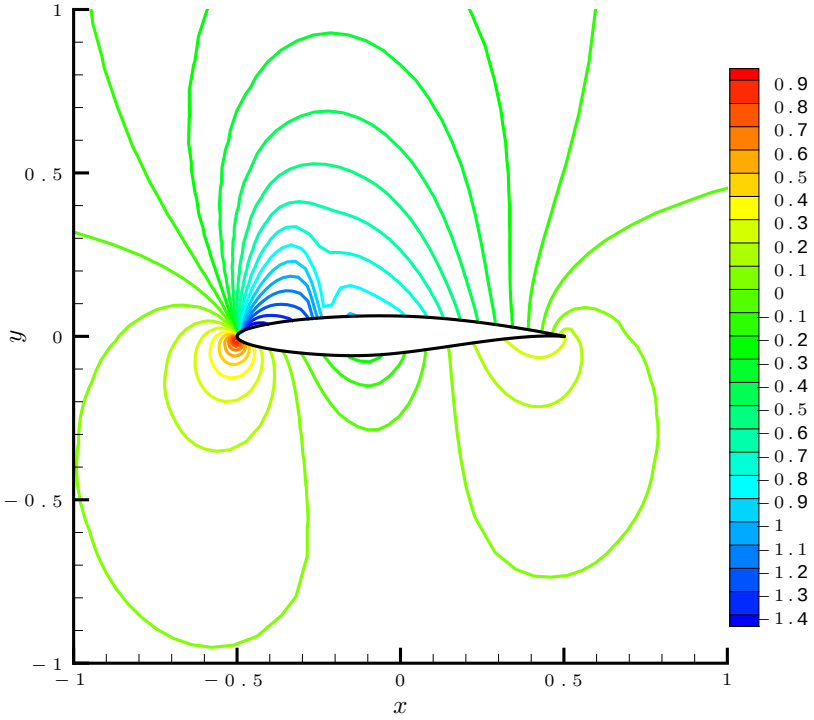
(a) $\boldsymbol{\mu}$ invariant methods

Figure 32: RMSE histories and iteration numbers as the number of data-missing snapshots changes

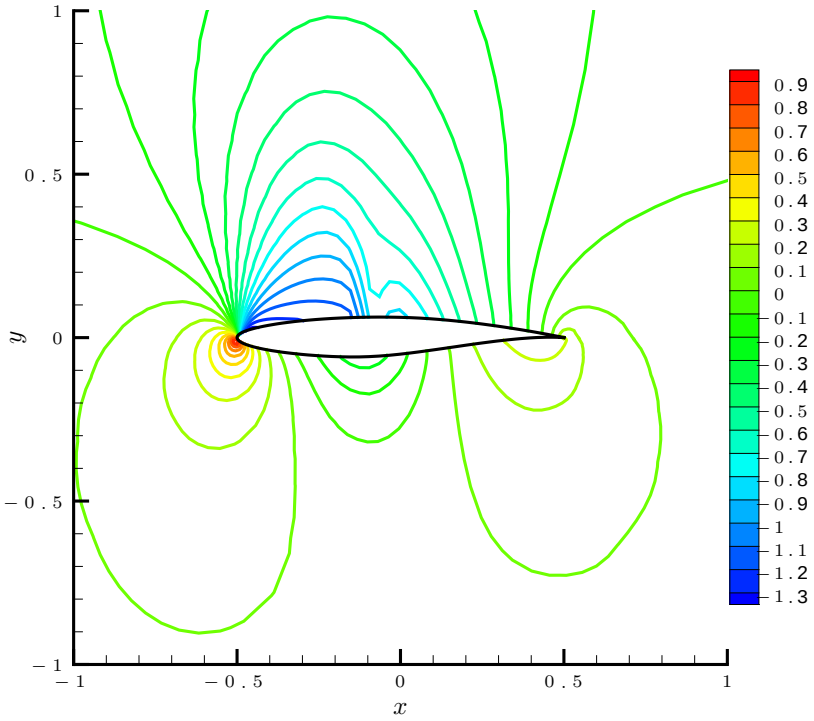


(b) μ variant methods

Figure 32: RMSE histories and iteration numbers as the number of data-missing snapshots changes



(a) C_p contours of the 24^{th} snapshot



(b) C_p contours of the 37^{th} snapshot

Figure 33: Examples of highly nonlinear snapshots due to local shock phenomena

computational time than $\mathbf{W}$. Thus, $\mathbf{W}$ is less numerically inexpensive than $\mathbf{V}_q$ for a single basis evaluation regardless of the size of q. Similarly, regarding a norm choice, at each iteration, the L^2 norm leads to a system of q linear equations for coefficient evaluations whereas the gappy norm results in a system of $q \times s$ linear equations proportional to s, the number of data-missing snapshots. As a result, the L^2 norm is more computationally advantageous than the gappy norm for obviating additional $q \times (s-1)$ linear equations. Thus, the computational benefit of the L^2 norm to the gappy norm hinges on not only the number of modes q but also the number of data-missing snapshots s.

Although $\mathbf{W}$ and the L^2 norm necessitate less computational effort per iteration than $\mathbf{V}_q$ and the gappy norm, the total number of iterations entailed by different bases and norms eventually determines the overall computational cost of the original and hybrid algorithms. As delineated in Section 4.3, the basis difference produces insignificant differentials in iteration numbers compared to the norm difference even though $\mathbf{V}_q$ usually requires fewer iterations than $\mathbf{W}$. Therefore, this research mainly focuses on addressing the norm effect on the convergence behavior of missing-data estimation algorithms. As an illustration, suppose that k_1 and k_2 represent the total number of iterations related to the gappy and L^2 norms, respectively. Since the gappy norm reduces more estimation errors per iteration than the L^2 norm as shown in Section 4.3, k_1 is observed less than k_2 such as $k_2 \approx 1 \sim 5k_1$ in numerical experiments tested with diverse missing data sets. The relationship between k_1 and k_2 depends on the amount of missing data as well as the nonlinearity of the data; the more missing data or nonlinearity, the wider the gap between k_1 and k_2. Under the observed most severe condition such that $k_2 = 5k_1$, the computational difference between the gappy and L^2 norms, evaluated as $q \times (sk_1 - k_2)$, becomes $q \times (s-5) \times k_1$.

All in all, provided that $k_2 \leq 5k_1$, algorithms using the gappy norm are anticipated to be faster than those using the L^2 norm as long as the number of data-missing snapshots s is less than five; otherwise, those using the L^2 norm are desirable for computational efficiency. Note that five is just a nominal number because the applications of missing data estimation algorithms are either of two cases: only one data-missing snapshot as in the first sample data or a numerous number of data-missing snapshots as in the second sample data.

The former mostly occurs when one attempts to exploit missing-data estimation methods to ingeniously address applications that do not belong to missing data estimation, such as flow data assimilation[84] or inverse airfoil design.[5] By contrast, the latter occurs when one utilizes missing data estimation methods such as PIV data restoration[57,58] and basis extraction from incomplete data[34] for which they were originally devised.

4.4.5 Formulation of Hypothesis 2.1

Through systematic comparative studies of the original and hybrid methods with artificial test data sets, this research is able to construct a hypothesis for Research Question 2.1.

Hypothesis 2.1. A norm selection affects estimation error reduction more than a basis selection.

For verification, Hypothesis 2.1 necessitates additional tests with different incomplete data sets. In this thesis, Hypothesis 2.1 will be partially evaluated for the second type of missing data set, which lacks data across an entire snapshot ensemble, in Chapter 7.

CHAPTER V

APPLICATION I: REDUCED-ORDER NPSS MODELING

5.1 Background

Historically, aircraft engine design has occurred in isolation of airframe design during the aircraft design processes. However, as advanced aircraft concepts such as NASA's N+1, N+2, and N+3 aircraft aim for technology levels in 10, 20, and 30 years, respectively, aircraft design research using the traditional engine design approach results in a limited design space. In order to broaden out a feasible design space for the next generation of aircraft, aircraft design must take into account not only airframe but also propulsion system aspects in parallel.[11,28] Since heuristic engine performance correlations traditionally used in aircraft design are inappropriate for such design studies, one requires a sophisticated physics-based engine simulation. For this purpose, the numerical propulsion system simulation (NPSS),[26,44,85] developed by the joint efforts of NASA and engine manufacturing industries, is a de facto standard tool in aerospace engineering. Due to its object-oriented architecture, NPSS provides a generic engine modeling framework that allows one to devise propulsion systems for various vehicles ranging from conventional configurations such as commercial jet transport, supersonic business jets, and military aircraft to visionary configurations such as hybrid wing body and truss-braced wing aircraft.

Although NPSS itself is not as computationally prohibitive as other high-fidelity, physics-based simulations, its capability to interface with high-fidelity codes can drastically increase its computational time on the order of hours. While the ROM of NPSS normally decreases computational time from minutes to seconds, it is still desirable for the following reasons: (i) Nondeterministic and optimization-driven design studies[48,49] repeatedly invoke NPSS a colossal number of times during a design process; (ii) directly interweaving NPSS into a multidisciplinary design environment may demand arduous integration work; and (iii) cooperative research entities that do not own NPSS can reap the benefits of NPSS analysis

capabilities through a reduced-order NPSS model. Particularly with regard to the first motivation, when an engine design becomes tightly combined with a vehicle design, aircraft system design efforts require that engine cycle parameters be varied in concert with vehicle design variables to arrive at a truly optimized, integrated design. As illustrated in Figure 34, this airframe- and engine-integrated design approach demands the repeated execution of NPSS, which becomes computationally expensive. Therefore, despite upfront computational cost, a reduced-order model of NPSS that can yield an engine deck almost instantaneously is indispensable to every aircraft system design process.

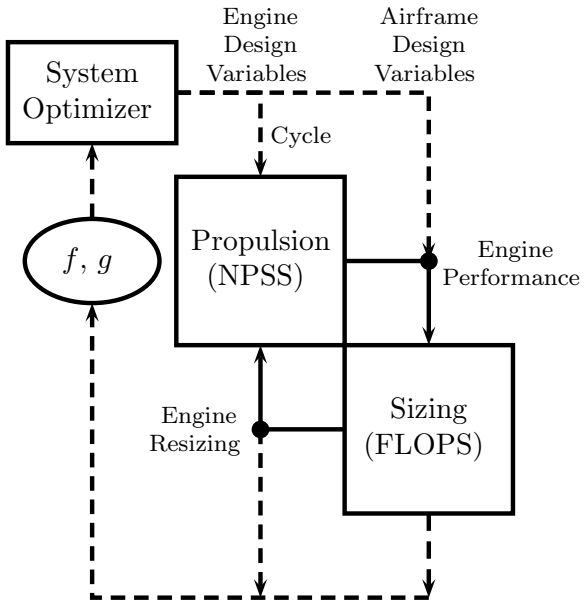


Figure 34: Schematic of an airframe- and engine-integrated design environment

For the modeling of engine performance metrics, semi-empirical equation-based approaches fitting measurement data through coefficient adjustments have been used to derive fuel consumption models;[71,88] however, these models are not generally accurate for all mission profiles and thus, they are limitedly acceptable for only certain mission segments. For example, Senzig, Fleming, and Iovinelli[71] pointed out that the fuel consumption model of base of aircraft data (BADA) is mostly accurate for a cruise mode, which inspired them to develop their fuel consumption model particularly accurate at climb/landing around terminal areas. In contrast to these restrictively applicable models, Lee et al.[34] attempted to devise generic prediction models of engine performance metrics with NPSS through its

reduced-order model. Conceptually, a reduced-order model is twofold: (i) an empirical orthogonal basis that is normally obtained by POD, referred to as PCA;[25,74] and (ii) weighting coefficients that adjust the contributions of basis vectors accordingly with changes in the independent modeling parameters.

To construct a reduced-order model of NPSS, one typically encounters research predicaments in both evaluating empirical orthogonal bases and determining weighting coefficients. Regarding empirical basis extraction, due to successive discontinuity shifts appearing at every throttle setting change, resultant NPSS engine decks contain distinct stationary discontinuities that lead to a sawtooth-type data pattern. Moreover, due to numerical instabilities occurring at certain off-design performance analyses within NPSS, some degree of data absence is inescapable with NPSS. Although an empirical orthogonal basis evaluated by POD can deal with variations in simulation data exhibiting stationary discontinuities,[41] POD fails for deficient data. In order to surmount the evaluation of a POD basis for given incomplete data such as NPSS engine decks, one can rely on either gappy POD[13] or PPCA;[82] while the former generates an orthogonal basis by solving a least-squares problem defined with a gappy norm and a POD basis, the latter yields a non-orthogonal basis through probability parameter estimation with the help of an EM algorithm for PPCA, termed an EM-PCA.

Since Bui-Thanh[4] showed the potential application of gappy POD to aerospace engineering problems, gappy POD has been utilized for various aerospace engineering applications such as inverse airfoil design[5] and spurious PIV data restoration.[58] Similarly, Lee, Rallabhandi, and Mavris[35] introduced PPCA to the realm of aerospace engineering; furthermore, Lee and Mavris[30,33] exhaustively investigated PPCA in comparison with gappy POD, demonstrating greater efficiency than gappy POD for PIV data reconstruction.[31] Unlike the previous applications of gappy POD and the EM-PCA, which focused on restoring missing data, Lee et al.[34] intended to exploit these methods to retrieve an empirical orthogonal basis from deficient simulation data. According to Lee and Mavris,[30] the EM-PCA is computationally superior to gappy POD for such a sparse missing data pattern of NPSS engine decks caused by occasionally failed off-design performance analyses. Therefore, Lee et al.[34] adopted the EM-PCA instead of gappy POD to efficiently distill the POD bases of

engine performance metrics from compiled NPSS engine decks.

The other research impediment to the general use of POD-based ROM for design applications is difficulties associated with predicting weighting coefficients as the number of independent modeling parameters grows. Once weighting coefficients at known independent modeling parameters are evaluated with the method of weighted residuals (MWR) [87] such as least-squares methods, one is required to infer weighting coefficients at unknown independent modeling parameters. In the case of POD-based ROM utilized for unsteady flow analysis that is time dependent only, the problem of estimating weighting coefficients is tractable; however, as POD-based ROM diffuses into design applications whose number of independent modeling parameters is typically more than one, weighting coefficient estimation turns into a matter of multivariate scattered data interpolation. [2]

For instance, Bui-Thanh, Damodaran, and Willcox [5] employed cubic spline interpolation in their reduced-order steady aerodynamic model to delineate variations in airfoil surface pressure due to two flow parameters: a Mach number and an angle of attack. Likewise, Mifsud, Shaw, and MacManus [52] utilized a POD-based reduced-order model for parametric studies of weapon aerodynamics, expanding the number of modeling parameters to three: a Mach number, an incidence angle, and a flare base radius. In addition, they tested a few DoEs, such as full factorial and Latin hypercube designs, and examined several response surface construction schemes including linear regression and polynomial augmented multi-quadratic RBF. Furthering these previous research endeavors, Lee et al. [34] capitalized on neural networks in connection with POD-base ROM to handle a large number of independent modeling parameters, namely six NPSS engine modeling parameters. Moreover, to ease the coefficient evaluation with neural networks, Lee et al. [34] benefited from an augmented DoE by adding the corner points of a parameter space to a maximum entropy design, a space-filling DoE specialized for computer simulations. [65,68]

Overall, this chapter aims to create a reduced-order NPSS model with the EM-PCA and neural networks, each of which is adopted to addresses inherently gappy NPSS engine decks and a large number of engine modeling parameters, respectively. The outline of this chapter is as follows. Section 5.2 describes the concept of the neural network, one of the two

rudimentary components for the proposed NPSS ROM method along with PPCA. Subsequently, Section 5.3 presents an NPSS engine model and the process of NPSS engine deck generation, followed by the results of prediction quality investigations that illustrate the accuracy of the reduced-order NPSS model compared to genuine NPSS. Section 5.4 completes this chapter with concluding remarks and recommendations for future work. Finally, Appendix B provides supplementary materials that show the EM-PCA yields identical results to those of gappy POD at lower computational cost, which substantiates the use of the EM-PCA over gappy POD.

5.2 Theories for NPSS Reduced-Order Modeling

5.2.1 Neural Networks

As an analogy to a biological neural network, an artificial neural network[12] is a computational architecture interconnecting artificial neurons to address various problems through artificial intelligence characterized as learning—in other words, generalization. Since neural networks are trained so that they can adapt to unexperienced situations, they have been utilized in diverse applications of data processing such as function approximation and data classification: image processing, engine management, automatic aircraft landing systems, and so on. As an illustration, Figure 35 depicts a simple feed-forward network with one hidden layer consisting of hidden nodes as similar to the neurons of biological neural networks. For a given training set, a neural network adjusts weights and biases between input and hidden layers as well as hidden and outer layers in order to match its outputs to known target values. For weight and bias determination, a back-propagation algorithm is commonly used in connection with a feed-forward network to minimize errors, differences between outputs and target values, such that it successively propagates errors backwards from output to input layers; for example, the weights and biases between the hidden and output layers are first rectified, and then those between hidden and input layers are modified.

For a single hidden-layer feed-forward network, depicted in Figure 35, the mathematical representation of neurons often relies on a sigmoid function

$$\sigma(t) = \frac{1}{1 + e^{-t}}, \tag{36}$$

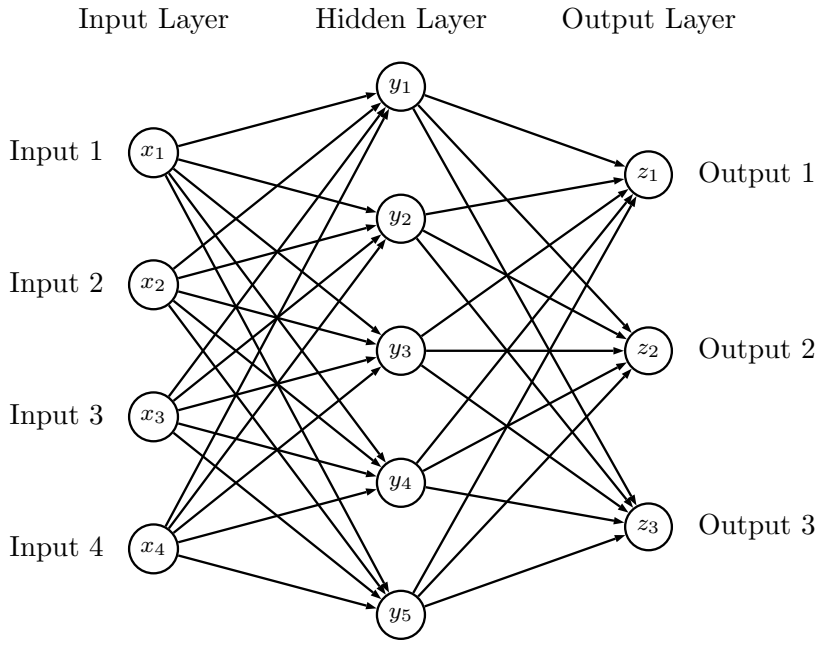


Figure 35: Single hidden-layer feed-forward neural network

which is one of the most widely-used activation functions.[78] Note that the sigmoid function in Eq. (36) represents gradual changes at input boundaries and rapid progression at in-between boundaries, squashing an input within the range of zero and one. Let $\mathbf{x} \in \mathbb{R}^l$, $\mathbf{y} \in \mathbb{R}^m$, and $\mathbf{z} \in \mathbb{R}^n$ be the inputs of neurons, the outputs of neurons, and the outputs of a single hidden-layer feed-forward network, respectively, shown in Figure 35; then a neuron output y_j determines its value with Eq. (36) as a function of a weighted sum of the input variables and a bias such that

$$y_j(x_i) = \sigma\left(\sum_{i=1}^{l} w_{ij}^{[x]} x_i + a_j^{[x]}\right),$$

where $w_{ij}^{[x]}$ is the weight of the i^{th} input variable x_i, and $a_j^{[x]}$ is the bias of the j^{th} hidden node. Similarly, the k^{th} output variable z_k is evaluated as

$$z_k(y_j) = \sum_{j=1}^{m} w_{jk}^{[y]} y_j + a_k^{[y]},$$

where $w_{jk}^{[y]}$ is the weight of the j^{th} hidden node, and $a_k^{[y]}$ is the bias of the k^{th} output variable. After all, a single hidden-layer feed-forward network assumes a nonlinear relationship

between the input and output variables such that

$$z_k(x_i) = \sum_{j=1}^{m} w_{jk}^{[y]} y_j + a_k^{[y]} = \sum_{j=1}^{m} w_{jk}^{[y]} \sigma \left(\sum_{i=1}^{l} w_{ij}^{[x]} x_i + a_j^{[x]} \right) + a_k^{[y]},$$

and $w_{ij}^{[x]}$, $w_{jk}^{[y]}$, $a_j^{[x]}$, and $a_k^{[y]}$ necessitate a training process to find their values minimizing errors.

5.2.2 POD-Based Reduced-Order Modeling

Given POD basis vectors $\{\mathbf{v}_j\}_{j=1}^{q} \in \mathbb{R}^{d \times q}$, $\mathbf{y} \in \mathbb{R}^d$ can be approximated on a q-dimensional subspace as

$$\mathbf{y}(\boldsymbol{\vartheta}, \mathbf{x}) \simeq \sum_{j=1}^{q} \alpha_j(\boldsymbol{\vartheta}) \mathbf{v}_j(\mathbf{x}) + \bar{\mathbf{y}}, \tag{37}$$

where $\bar{\mathbf{y}}$ is a sample mean, and α_j is the optimal weighting coefficient of $\mathbf{v}_j$, minimizing a projection error. Note that POD basis vectors account for spatial variations in $\mathbf{y}$ due to changes in $\mathbf{x}$, and likewise, the coefficients do so for the parametric variations in $\mathbf{y}$ due to changes in $\boldsymbol{\vartheta}$. Provided that POD basis vectors are invariant to $\boldsymbol{\vartheta}$, POD-base ROM can predict $\mathbf{y}$ at unobserved parameters in a form similar to Eq. (37) such that

$$\widetilde{\mathbf{y}}(\boldsymbol{\vartheta}, \mathbf{x}) \approx \sum_{j=1}^{q} \widetilde{\alpha}_j(\boldsymbol{\vartheta}) \mathbf{v}_j(\mathbf{x}) + \bar{\mathbf{y}}, \tag{38}$$

with the help of an appropriate weighting coefficient $\widetilde{\alpha}_j$. For the evaluation of POD basis vectors and weighting coefficients, this research employs the EM-PCA and neural networks, respectively, to construct a reduced-order model of NPSS.

5.3 Generation of a Reduced-Order NPSS Model

A two-spool, separate flow turbofan was created as a thermodynamic cycle model within NPSS using the typical flight envelop of a commercial jet aircraft. To generate engine performance information, NPSS normally runs in two stages; the first design analysis sizes engine components according to desired engine cycle parameters, and the second off-design analysis, called a performance analysis, produces engine performance data at each set of engine operating conditions consisting of a Mach number, an altitude, and a throttle setting.* As shown in Table 6, NPSS provides the results of an engine performance analysis

*A throttle setting is also known as a power code or a power-level angle.[66]

as an engine deck that adjoins two tables; the table of engine deck inputs on the left lists various combinations of engine operating conditions, and the table of engine deck responses on the right arranges corresponding engine performance metrics such as gross thrust, ram drag, fuel flow, and so forth.

Table 6: NPSS engine deck format

Engine deck inputs			Engine deck responses		
Mach number	Altitude [ft.]	Throttle setting	Gross thrust [lbf]	Ram drag [lbf]	Fuel flow [lbm/hr]
0	0	50	90,000	0	27,000
⋮	⋮	⋮	⋮	⋮	⋮
0.9	43,000	21	14,162	15,150	520.5

In general, a functional relationship between engine deck inputs and responses hinges on the following factors: an engine cycle, an engine architecture, and the underlying fidelity of engine component models. Among these three elements, the architecture of an engine is often chosen first in conceptual aircraft design based on the mission requirements of an aircraft as well as the financial wherewithal and competitive assessment of an engine manufacturer. Once an engine architecture is determined, one manipulates the engine cycle parameters so that a designed engine can perform to the level of the mission requirements imposed on a vehicle of interest. According to Schobeiri,[69] engine cycle changes give rise to many secondary effects, such as the influence of component pressure ratios on component efficiencies, and most of these subsidiary effects are included in the NPSS model, implemented for this research.

5.3.1 NPSS ROM Procedure

The following steps briefly describe the construction of a reduced-order NPSS model with the help of EM-PCA and neural networks. More details regarding each step will be presented accordingly after this subsection.

Step 1 Generate engine deck snapshots using NPSS at various engine cycle parameters.

This step populates sample engine decks not only for extracting empirical orthogonal bases with the EM-PCA but also for training weighting coefficient models with neural networks. One can benefit from DoE techniques specialized for computer simulations to effectively capture an entire engine modeling parameter space. Note that a DoE is required to encompass the corners of the parameter space in order for a reduced-order model to properly account for all variations within the parameter space.

Step 2 Compile each engine deck response of interest.

This step separately collects engine responses from populated engine decks, producing snapshot ensembles of engine deck responses for basis extraction with the EM-PCA.

Step 3 Utilize the EM-PCA to obtain the empirical orthogonal bases of engine deck responses.

This step capitalizes on the EM-PCA to distill empirical orthogonal bases from the snapshot ensembles of engine deck responses. Based on eigenvalues representing the relative contributions of corresponding basis vectors, one is required to choose a proper number of basis vectors that sufficiently delineate all variations in the snapshot ensembles. Note that the EM-PCA technically yields a non-orthogonal basis that necessitates orthogonalization for an orthogonal basis.

Step 4 Evaluate optimal weighting coefficients by least-squares methods.

This step determines the best coefficients, those that produce minimum projection errors for given empirical orthogonal bases. Owing to the orthogonality of the empirical bases, the evaluation of weighting coefficients reduces to mere matrix multiplication dispensing with matrix inversion. The optimal coefficients achieved in this step will be targets for neural network models in Step 5.

Step 5 Utilize neural networks to build the prediction models of weighting coefficients.

This step constructs coefficient prediction models based on function-approximation

neural networks to estimate weighting coefficients at unseen engine modeling parameters. For the prediction models of weighting coefficients, the engine modeling parameters in the DoE table in Step 1 are taken as inputs, and the optimal weighting coefficients evaluated in Step 4 are fed as targets. Note that the coefficient prediction models significantly sway the goodness of a reduced-order NPSS model since weighting coefficients are the only factors that can account for changes in engine modeling parameters; empirical bases are assumed to be invariant to the parameter changes in the formulation of POD-based ROM.

Step 6 Examine model-fit-errors with training data.

This step carries out model-fit-error tests for evaluating the performance of the reduced-order NPSS model built upon the empirical bases in Step 3 and the neural network models in Step 5. Although the optimal weighting coefficients achieved in Step 4 can replicate observed engine deck responses, weighting coefficients estimated by the neural network models in Step 5 do not necessarily yield the exact weighting coefficients at known engine modeling parameters. Therefore, model-fit-error tests are useful for validating the quality of the prediction models of weighting coefficients.

Step 7 Examine model prediction errors with randomly-generated test data.

This final step repeats the same error analysis in Step 6 with random test data that are new to the reduced-order NPSS model. This prediction test with random data is conducive to examining the overall performance of the reduced-order model at unobserved engine modeling parameters.

5.3.2 NPSS Engine Deck Generation

To construct a reduced-order NPSS model, this research varied the following six engine modeling parameters: an extraction ratio, a fan pressure ratio (FPR), a high-pressure compressor pressure ratio (HPCPR), a low-pressure compressor pressure ratio (LPCPR), a maximum turbine inlet temperature (MaxT41), and sea-level static thrust (SLST). These

engine cycle and scaling parameters are commonly utilized in conceptual engine design.[47] Each parameter has its own unique impact on not only engine performance but also the closeness of engine performance to thrust and fuel flow requirements. Overall, different combinations of the six parameters, whose ranges are presented in Table 7, populated a total of 500 sample engine decks. For the strategic exploration of an engine modeling parameter space, the generation of sample engine decks adopted a maximum-entropy space-filling design[24] augmented with a total of 76 corner points using JMP software. At each set of the engine modeling parameters, NPSS produced engine performance data with respect to 924 engine operating conditions composed of 11 altitudes, $5 \sim 10$ Mach numbers, and 11 throttle settings. As an illustration, Table 8 shows various combinations of engine operating altitudes and Mach numbers employed for the NPSS analysis in this research. Note that each altitude is associated with dissimilar sets of Mach numbers even though a constant throttle setting is related with a Mach number such that $(50, 48, 46, 42, 38, 34, 30, 26, 24, 22, 21)$; 50 and 20 represent maximum and idle throttles, respectively.[66]

Table 7: Ranges of the NPSS engine cycle and scaling parameters

	Extraction ratio	FPR	HPCPR	LPCPR	MaxT41 [°R]	SLST [lbf]
Minimum	1	1.5	18	1.2	3200	$80,000$
Maximum	1.2	1.7	22	1.6	3600	$100,000$

As a result, a total of 500 snapshots of an engine deck are generated and compiled, resulting in four 924-by-500 snapshot ensembles for four different engine deck responses of interest in conceptual aircraft design: gross thrust, ram drag, fuel flow, and emission index NO_X (EINOx). These snapshot ensembles inevitability lack some off-design performance analyses at certain flight conditions since the Newton–Raphson method in NPSS sporadically causes convergence failures leading to absent performance information, particularly at low throttle settings. As an illustration, Figure 36 depicts the locations of failed performance analyses observed in the collected snapshots of engine deck responses. As shown in Figure 36, a training data set is deficient of 0.143% data in Figure 36(a), and a test data

Table 8: Ranges of operating Mach numbers associated with altitude changes

Altitude [ft.]	0	2,000	5,000	10,000	15,000	20,000	25,000	30,000	35,000	39,000	43,000
Mach Number	0.0	0.0	0.0	0.1	0.3	0.4	0.45	0.55	0.6	0.6	0.7
	0.1	0.1	0.1	0.2	0.35	0.45	0.5	0.6	0.65	0.65	0.75
	0.2	0.2	0.2	0.25	0.4	0.5	0.55	0.65	0.7	0.7	0.8
	0.25	0.25	0.25	0.3	0.45	0.55	0.6	0.7	0.75	0.75	0.85
	0.3	0.3	0.3	0.35	0.5	0.6	0.65	0.75	0.8	0.8	0.9
	0.35	0.35	0.35	0.4	0.55	0.65	0.7	0.8	0.85	0.85	
		0.4	0.4	0.45	0.6	0.7	0.75	0.85	0.9	0.9	
			0.45	0.5	0.65	0.75	0.8				
				0.55	0.7	0.8	0.85				
						0.85					

set is absent of 0.12% data in Figure 36(b). In addition, Figure 37 shows the number of failed performance analyses per engine deck for both the training and test data sets; the maximum numbers of failed performance analyses for the training and test data sets are five and six out of 924 performance analyses, respectively. After all, despite a relatively minute percentage of data deficiency, POD is incapable of dealing with the snapshot ensembles of engine deck responses containing missing data; hence, the EM-PCA is required to extract empirical orthogonal bases from the training data set.

5.3.3 Implementation of a Reduced-Order NPSS Model

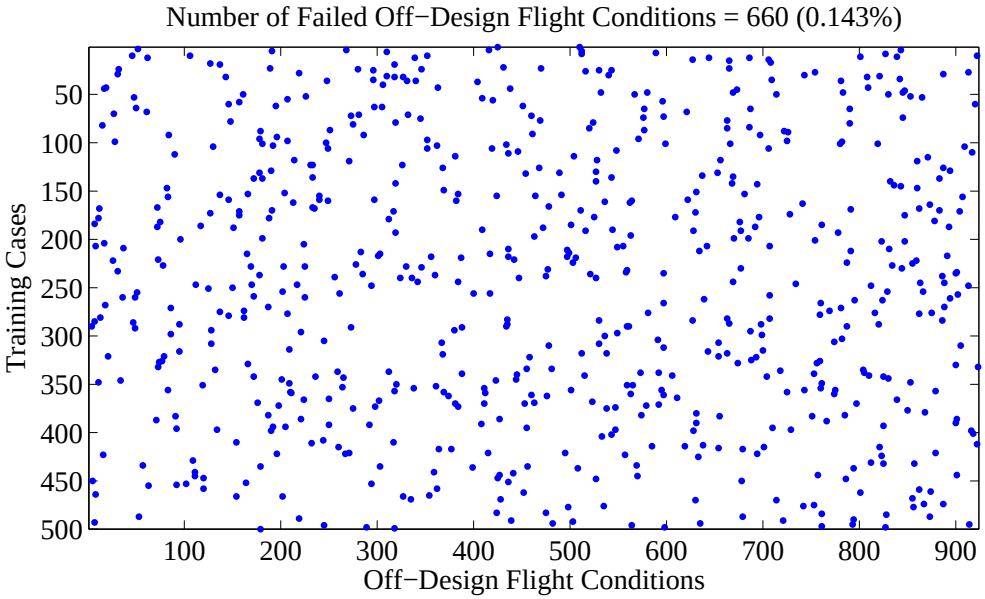
5.3.3.1 EM-PCA for Basis Extraction

The EM-PCA is implemented in MATLAB with two algorithmic variations: (i) whether to evaluate a sample mean and update a mean-centered snapshot ensemble at each iteration, and (ii) how to initialize a non-orthogonal basis $\mathbf{W}$ before the onset of iterations. For notational convenience, the former is denoted by "$\boldsymbol{\mu}$ inv."/"$\boldsymbol{\mu}$ var." and the latter by "`rand`"/"$\mathbf{V}_e$" in the names of EM-PCA implementations. For instance, "$\boldsymbol{\mu}$ inv." indicates that an EM-PCA implementation treats both a sample mean and a mean-centered snapshot ensemble as constant during iterations. In contrast, "$\boldsymbol{\mu}$ var." implies the other case, which allows an EM-PCA implementation to vary both a sample mean and a mean-centered snapshot per iteration. Similarly, "`rand`" shows that an EM-PCA implementation initializes $\mathbf{W}$ with a random matrix whereas "$\mathbf{V}_e$" expresses that an EM-PCA implementation initializes $\mathbf{W}$ with a guessed POD basis $\mathbf{V}_e$ obtained from an estimated snapshot ensemble whose missing data are filled with a sample mean beforehand to initiate iterations. As a result, a total of four EM-PCA implementations are generated, and an RMSR such that

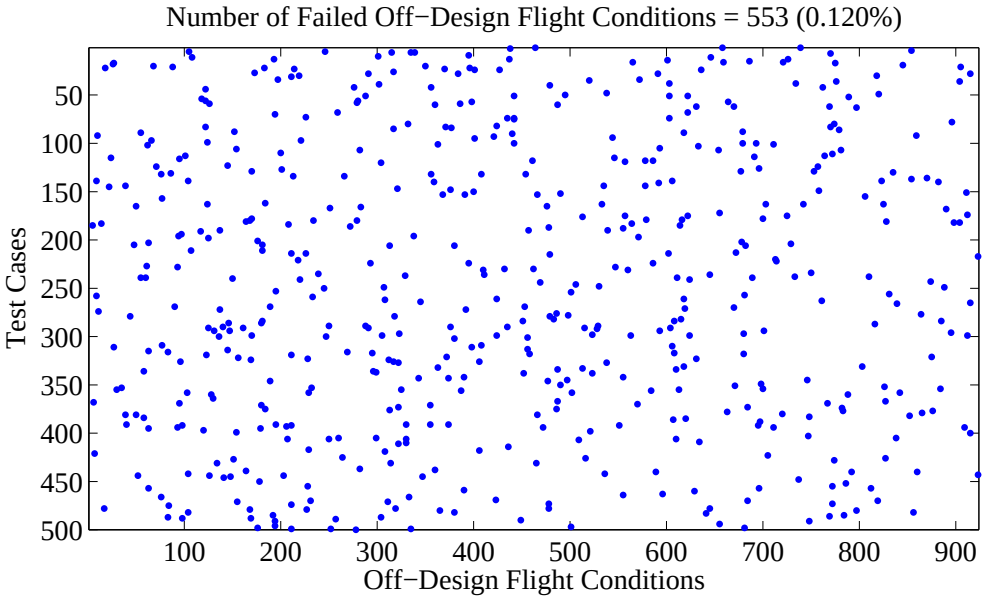
$$\mathrm{RMSR}^{(k)} = \sqrt{\frac{1}{dN}\sum_{j=1}^{N}\left\|\widetilde{\mathbf{y}}_j^{(k)} - \widetilde{\mathbf{y}}_j^{(k-1)}\right\|_{L^2}^2} \tag{39}$$

is employed to determine their convergence in terms of a normalized RMSR with respect to the first RMSR. For a convergence threshold, a normalized RMSR is set to 10^{-6} for all EM-PCA implementations.

For the given snapshot ensembles of engine deck responses, all four EM-PCA implementations yield virtually identical eigenspectra, modes, and restored failed performance

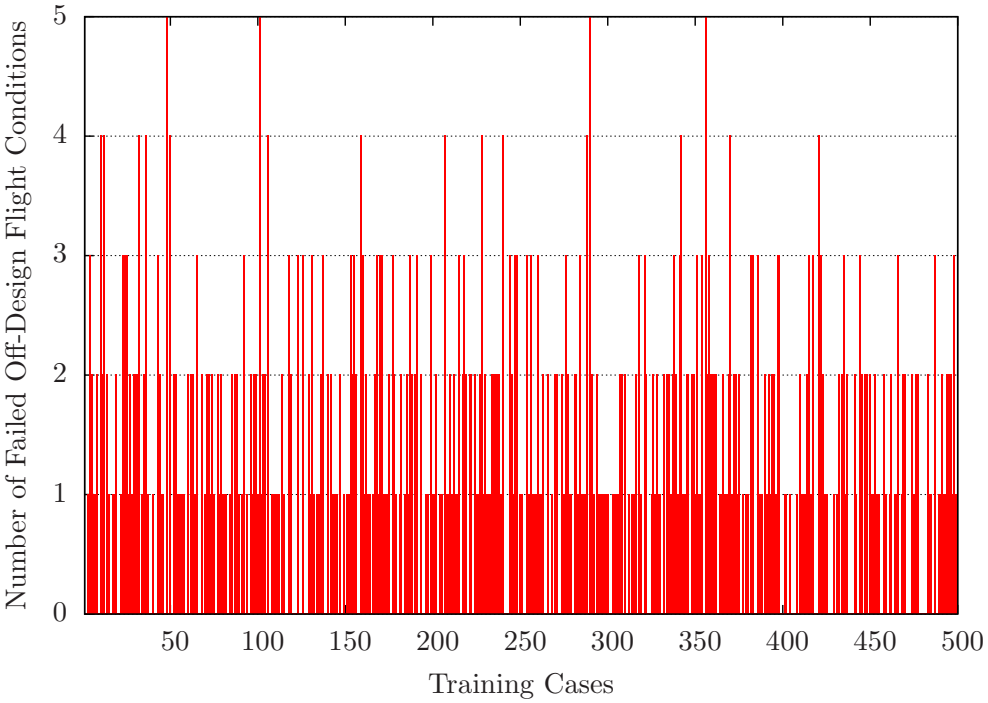


(a) Training data

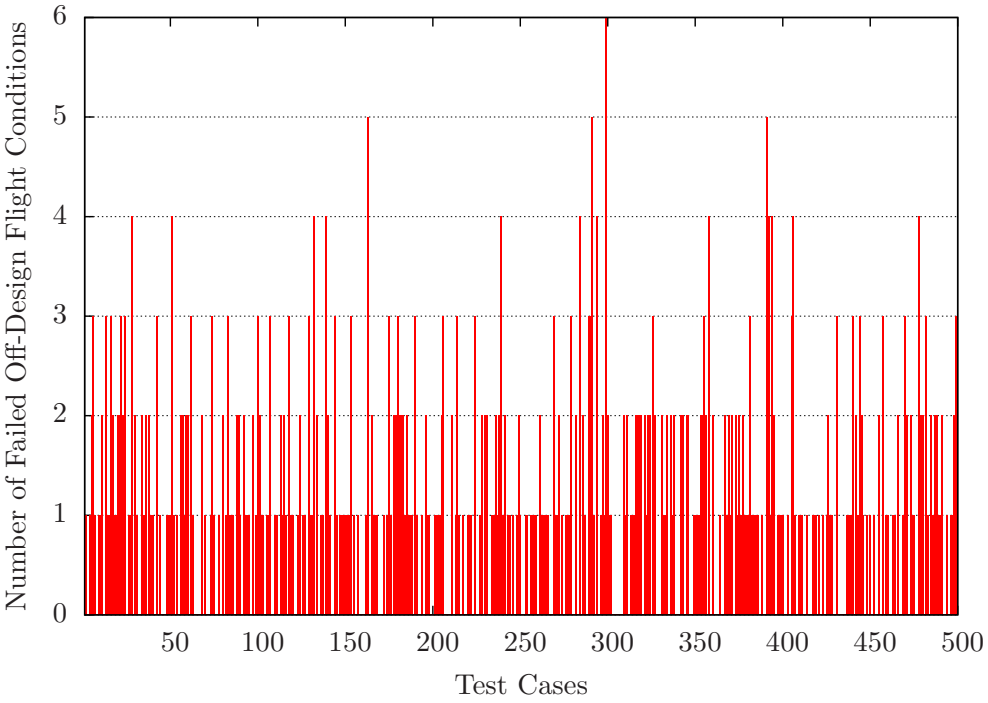


(b) Test data

Figure 36: Distributions of failed NPSS off-design performance analyses



(a) Training data



(b) Test data

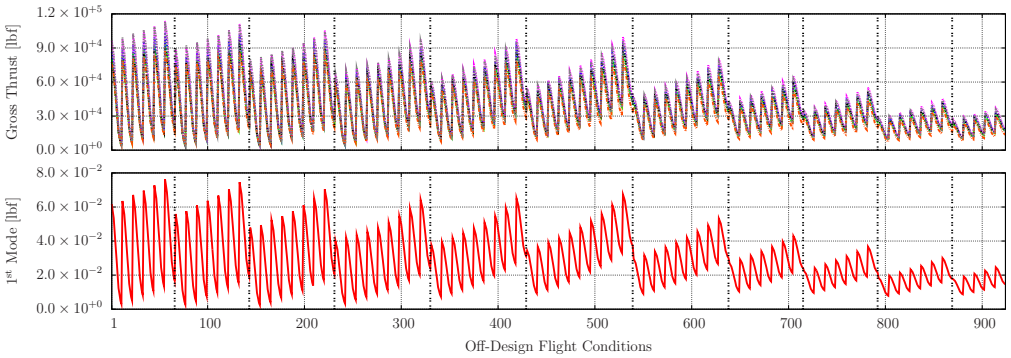
Figure 37: Number of failed NPSS off-design flight analyses

analyses. As an illustration, Table 9 shows the eigenspectra of engine deck responses, evaluated by the EM-PCA $\boldsymbol{\mu}$ inv.: `rand`, and these normalized eigenvalues with respect to their sum generally indicate relative variations associated with corresponding modes in engine deck responses. Overall, just four modes are found to be sufficient enough to account for 99.9% of variations in all four snapshot ensembles of the engine deck responses because of their little fluctuations with respect to the engine modeling parameters in Table 7. Therefore, significant dimensionality reduction from 924 to four is achieved for the four engine deck responses by virtue of the EM-PCA implementations. As an illustration of the obtained modes, Figure 38 shows engine deck responses sampled at every 100^{th} snapshot along with their first modes. Since the first modes are considerably dominant, as noted by their normalized eigenvalues over 0.9 in Table 9, the first modes clearly capture the most general behavior of the engine deck responses depicted in Figure 38. After all, this research utilizes four modes to develop a reduced-order model of the four engine deck responses based on the eigenspectrum analysis in Table 9.

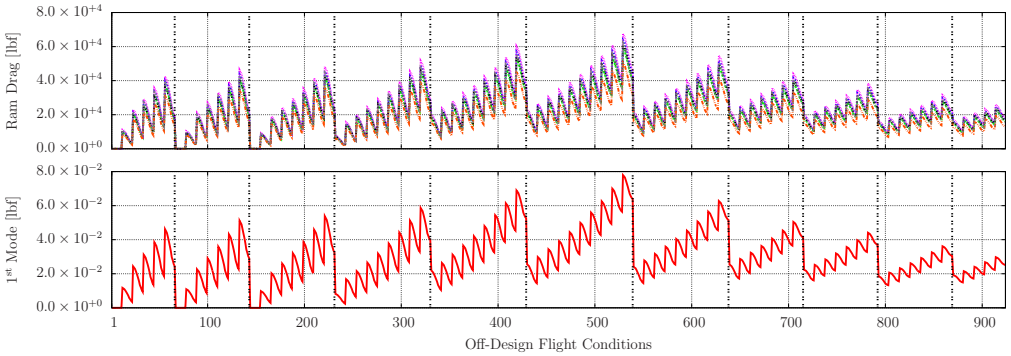
Table 9: Normalized eigenspectra of engine deck responses

	Gross thrust	Ram drag	Fuel flow	EINO_X
λ_1	$9.016077e{-01}$	$9.884888e{-01}$	$9.187537e{-01}$	$9.702735e{-01}$
λ_2	$7.471218e{-02}$	$8.252409e{-03}$	$7.289498e{-02}$	$2.574894e{-02}$
λ_3	$2.179765e{-02}$	$3.052206e{-03}$	$4.010828e{-03}$	$1.988086e{-03}$
λ_4	$1.453996e{-03}$	$1.425448e{-04}$	$3.268362e{-03}$	$1.155463e{-03}$
$\sum_{j=1}^{4} \lambda_j$	$9.995715e{-01}$	$9.999360e{-01}$	$9.989278e{-01}$	$9.991660e{-01}$

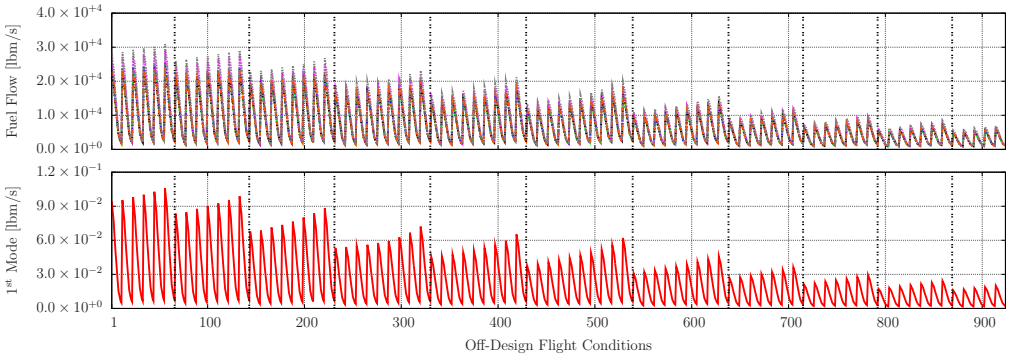
Figure 39 delineates the convergence histories of the four EM-PCA implementations using four modes for the given snapshot ensembles of engine deck responses. In Figure 39, all EM-PCA implementations quickly reach their convergence thresholds, owing to the insignificant amount of missing data in the snapshot ensembles of the engine deck responses. Note that dissimilar $\mathbf{W}$ initializations clearly differentiate the convergence behavior of the EM-PCA implementations in Figure 39; the EM-PCA implementations with $\mathbf{V}_e$ require fewer iterations than those with `rand` regardless of the $\boldsymbol{\mu}$ var. and $\boldsymbol{\mu}$ inv. implementations.



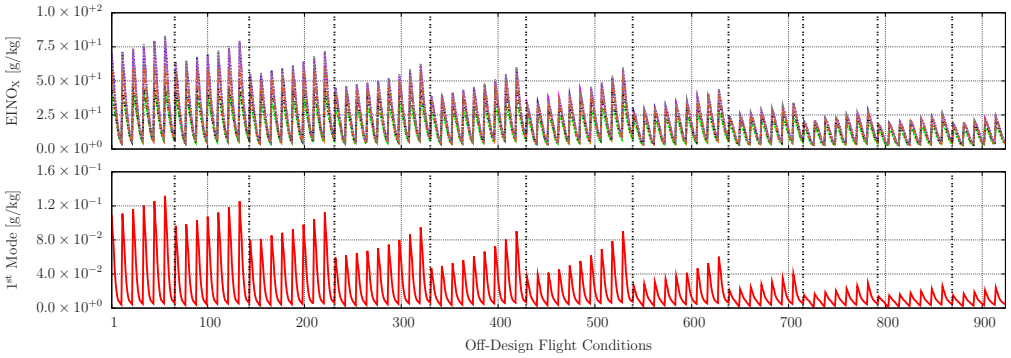
(a) Gross thrust



(b) Ram drag



(c) Fuel flow



(d) $\mathrm{EINO_X}$

Figure 38: Sampled snapshots of engine deck responses with their first modes

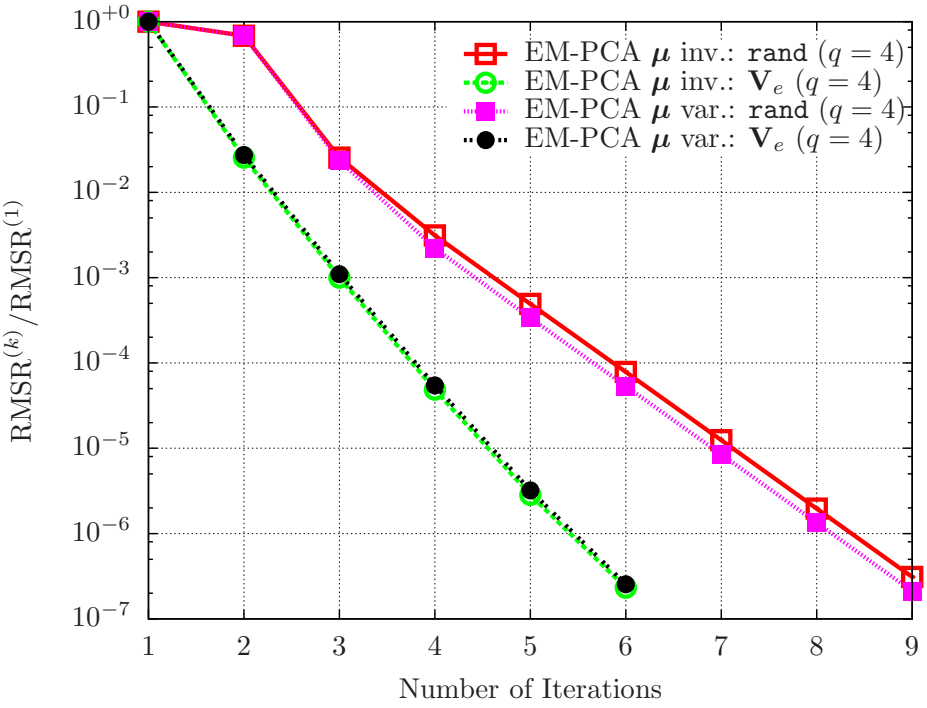
This convergence advantage of the $\mathbf{W}$ initialization with $\mathbf{V}_e$ mainly results from the low percentage of missing data and the uncomplicated variations of compiled engine deck responses, both of which are conducive to obtaining $\mathbf{V}_e$ close to the true $\mathbf{V}_q$. Unlike the $\mathbf{W}$ initialization difference in terms of `rand` and $\mathbf{V}_e$, the $\boldsymbol{\mu}$ inv. and $\boldsymbol{\mu}$ var. difference produces almost no influence on the convergence histories of the EM-PCA implementations, which implies that a sample mean fluctuates very little during iterations.

5.3.3.2 Neural Networks for Coefficient Fitting

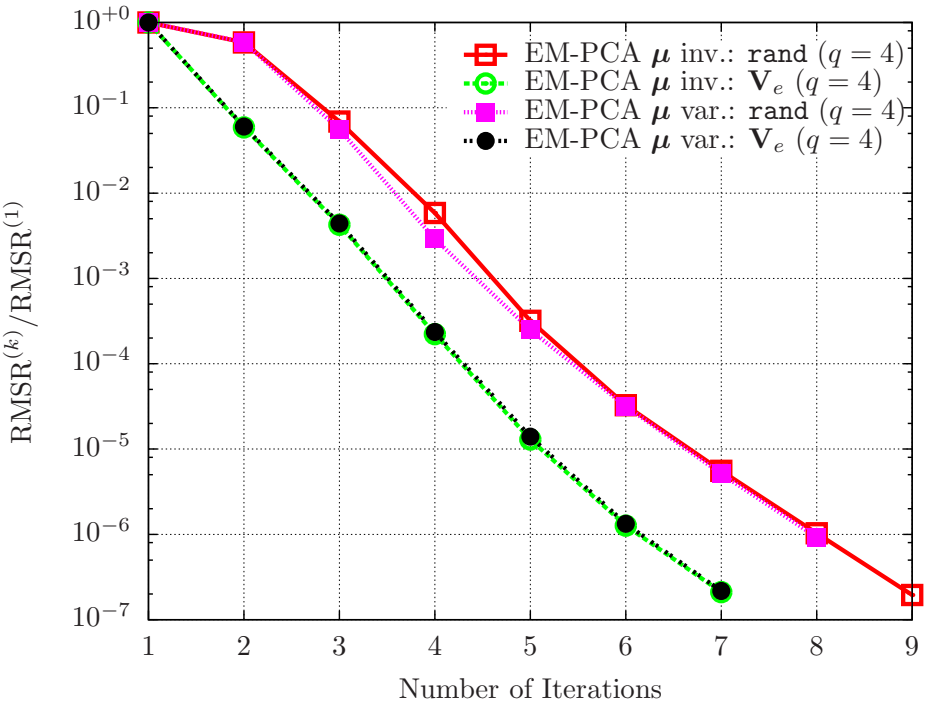
In order to build coefficient prediction models through neural networks, this research utilizes `newff` in MATLAB, producing feed-forward network models consisting of a single hidden-layer with 40 neurons. For a neuron activation function, the neural network models adopt a typical tan-sigmoid transfer function in Eq. (36) implemented by `tansig`, and for a back propagation algorithm, the Levenberg–Marquardt method realized by `trainlm` is invoked with the following stopping rules: the maximum number of epoches is 100 and the convergence threshold is 10^{-6} in terms of a mean square error (MSE). Since the magnitude of weighting coefficients is considerably larger than that of normalized empirical bases, training processes are mostly terminated by reaching the maximum number of epoches. In the training process, each randomly chosen 20% of training data is reserved for validation and testing, and then full training data are used to fit the final coefficient prediction modes. Overall, Table 12 presents the quality of weighting coefficient models in terms of the coefficient of determination (R^2), demonstrating quite good fitting results.

Table 10: R^2 of the weighting coefficients of engine deck responses for training data

	Gross thrust	Ram drag	Fuel flow	EINO_X
α_1	$9.999979e{-}01$	$9.999985e{-}01$	$9.999984e{-}01$	$9.999989e{-}01$
α_2	$9.999865e{-}01$	$9.999432e{-}01$	$9.999871e{-}01$	$9.999603e{-}01$
α_3	$9.999564e{-}01$	$9.998302e{-}01$	$9.998875e{-}01$	$9.998023e{-}01$
α_4	$9.986888e{-}01$	$9.952442e{-}01$	$9.995231e{-}01$	$9.990423e{-}01$

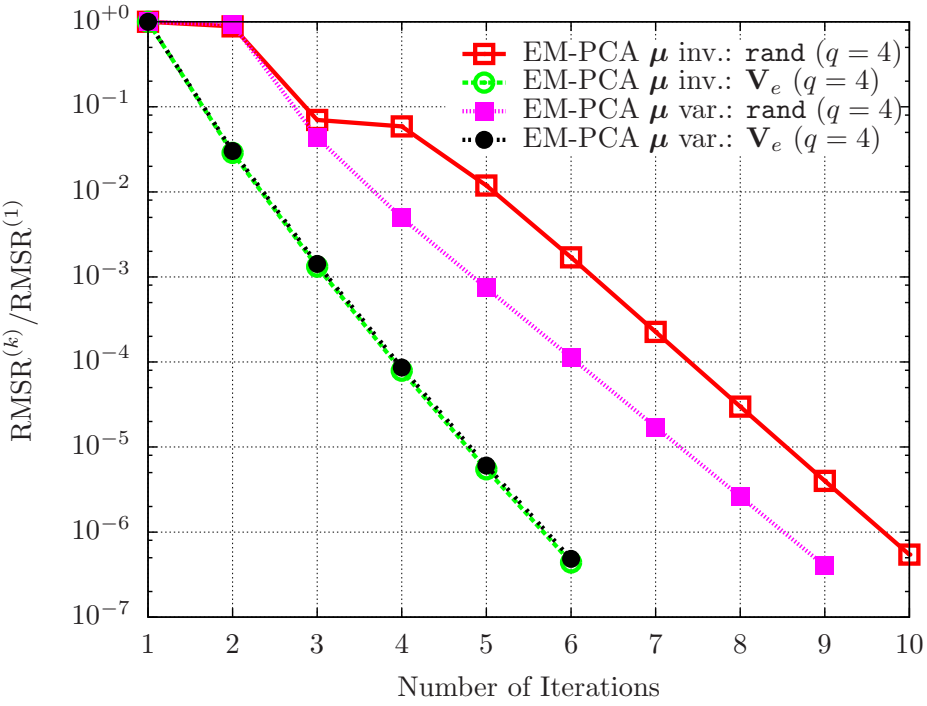


(a) Gross thrust

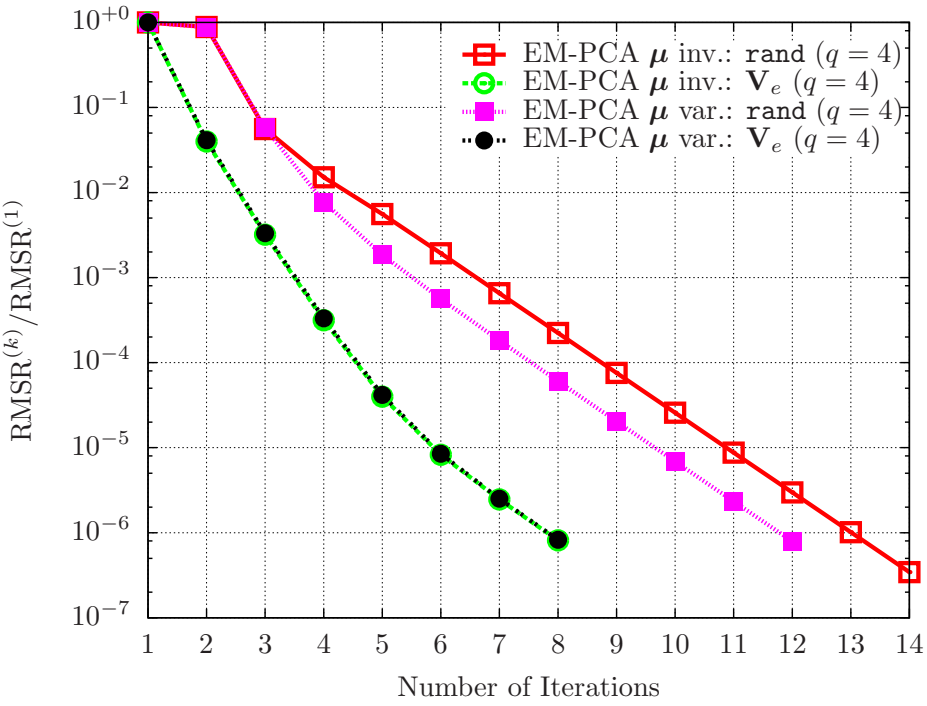


(b) Ram drag

Figure 39: Convergence histories of EM-PCA implementations



(c) Fuel flow



(d) EINO_X

Figure 39: Convergence histories of EM-PCA implementations

5.3.4 Goodness-of-Fit Analysis

5.3.4.1 Model-Fit-Error Test

As the first step toward validating the previously developed reduced-order NPSS model via the EM-PCA and neural networks, this research examines the performance of the reduced-order model with the training data. Since the empirical bases of engine deck responses are truly invariant to known training data, the prediction quality of the reduced-order model hinges solely on neural network models for weighting coefficient evaluations. For the quantification of the prediction capability of the reduced-order model, the following three metrics are employed: an R^2, a normalized root square error (NRSE) such that

$$\text{NRSE} = \sqrt{\left\| \frac{n_{ij}y_{ij} - n_{ij}\widetilde{y}_{ij}}{n_{ij}y_{ij}} \right\|_{L^2}^2}, \tag{40}$$

and a normalized root mean square error (NRMSE) defined by

$$\text{NRMSE} = \sqrt{\frac{1}{\sum_{i=1}^{d} n_{ij}} \sum_{j=1}^{N} \left\| \frac{\mathbf{n}_j \circ \mathbf{y}_j - \mathbf{n}_j \circ \widetilde{\mathbf{y}}_j}{\mathbf{n}_j \circ \mathbf{y}_j} \right\|_{L^2}^2}. \tag{41}$$

In Eqs. (40) and (41), $\circ$ denotes point-wise multiplication, and $\mathbf{n}_j \in \mathbb{R}^d$ indicates the existence of the elements of $\mathbf{y}_j$ as follows:

$$n_{ij} = \begin{cases} 0 & \text{if } y_{ij} \text{ is missing,} \\ 1 & \text{if } y_{ij} \text{ is known,} \end{cases} \quad \text{for} \quad i = 1, \ldots, d.$$

Because of occasionally absent engine deck responses in failed performance analyses, both NRSE and NRMSE require $\mathbf{n}_j$ in their evaluations to ignore unavailable data. Note that NRSE and NRMSE are normalized with respect to a true value, implying a relative error of a predicted value compared to the true value. In view of error analysis, an R^2 represents the overall fitness of predicted values to exact values; however, both NRSE and NRMSE quantify the deviations of predicted values from true values. Specifically, an NRSE measures the relative difference between a predicted and an exact value, and an NRMSE is the average of NRSEs for an entire engine deck response. For the training data, R^2 values are at least over 0.9975 in Figure 40, demonstrating the exceptional prediction capability of the reduced-order NPSS model. In Figure 41, NRMSEs are mostly less than 2% and increase to around

6% in Figure 41(a); similarly, maximum NRSEs are mostly less than 5% and rise up to around 30% in Figure 41(b).

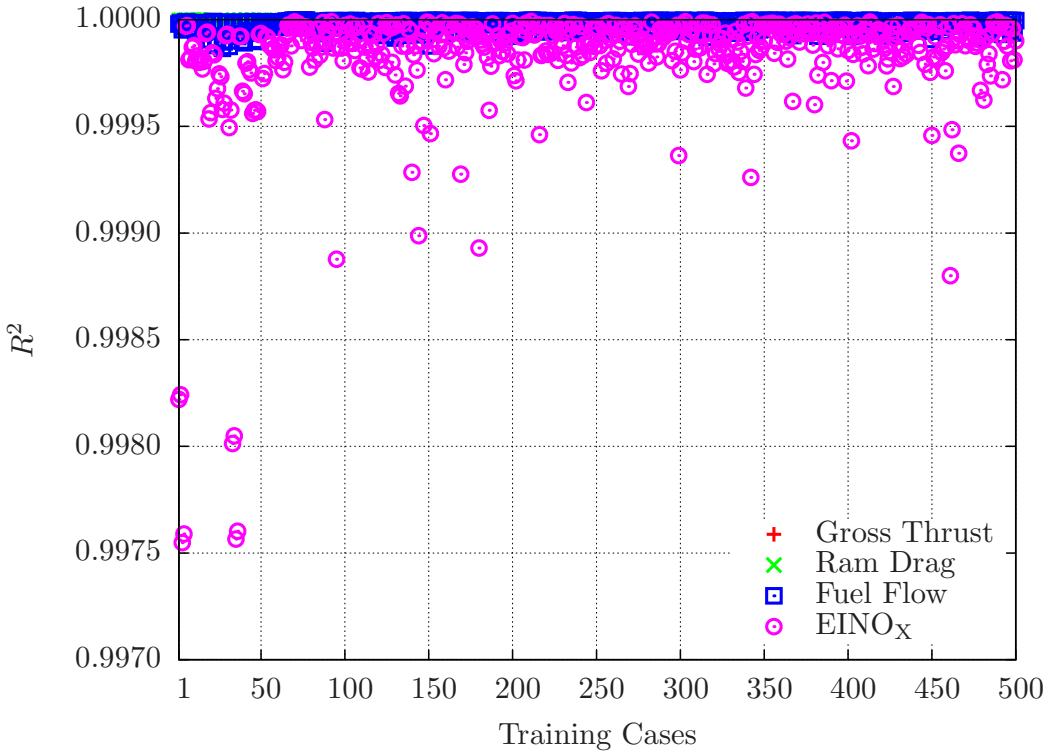


Figure 40: R^2 for training data

Although Figure 41(b) exhibits some training cases with large maximum NRSEs, these maximum NRSEs occur only in a single particular off-design flight condition among the 924 off-design flight conditions. Therefore, certain training cases with high maximum NRSEs in Figure 41(b) do not necessarily entail worse prediction results in terms of R^2 and NRMSE. For instance, although the highest NRSE for EINOx is 26.05% in the 434^{th} training case in Figure 41(b), the overall prediction quality of the case is not inferior at all, as indicated by its R^2 and NRMSE of 0.9999162 and 1.31%, respectively. Moreover, relatively high maximum NRSEs in Figure 41(b) are mostly caused by the numerical instability of NPSS, which will be discussed in detail later in Section 5.3.4.3. Overall, the predicted accuracy of the four engine deck responses, from most to least accurate, were gross thrust, ram drag are, fuel flow, and EINOx. Note that using the neural network models, one can reproduce engine deck responses in the training data with available optimal least-squares coefficients

instead of estimated weighting coefficients.

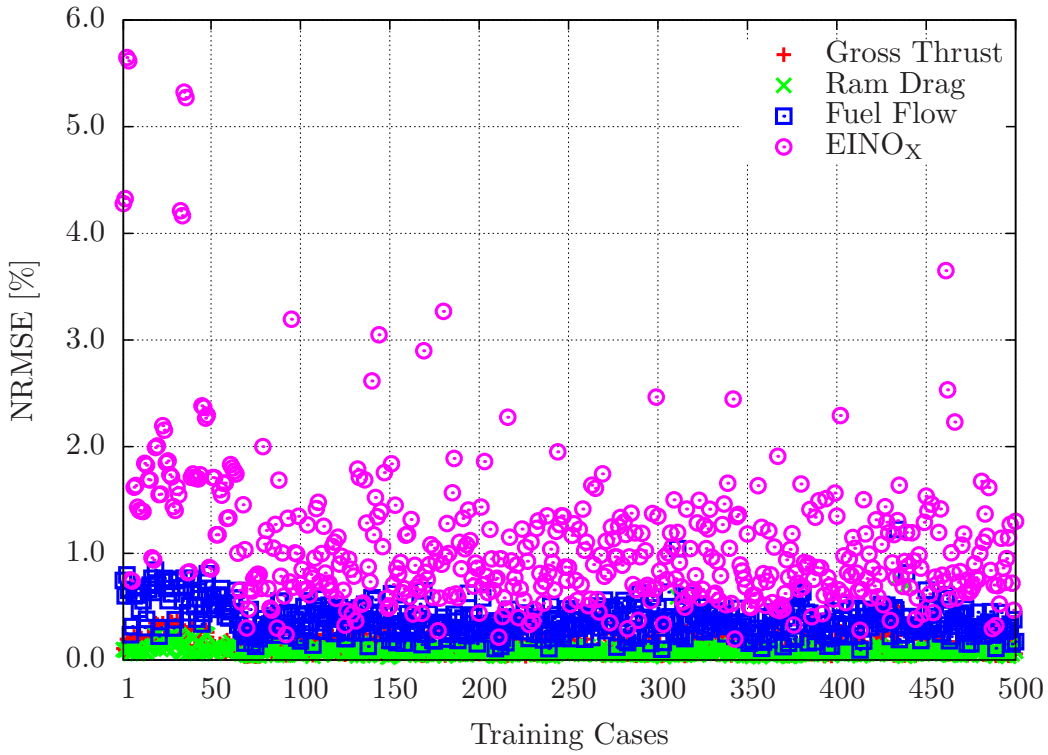
5.3.4.2 Quality of Empirical Bases and Coefficients

In order to delve into the source of prediction errors, this research examines the quality of both the empirical bases and the weighting coefficient models for the random test data. First, with regard to the empirical bases, this research measures the differences between the empirical bases of the training data and those of the test data in terms of an NRMSE. Table 11 shows that the changes in the empirical bases with respect to the two disparate data sets are insignificant, and the order of magnitude is quite similar across the four engine deck responses; the most dominant first modes exhibit variations one order smaller than the other three subordinate modes. Note that the empirical bases of the test data do not necessarily represent the true bases of the engine deck responses because the test data are randomly scattered over the space of the engine modeling parameters. Overall, Table 11 sufficiently corroborates the assumed invariancy of the empirical bases for the sample data space of the engine modeling parameters.

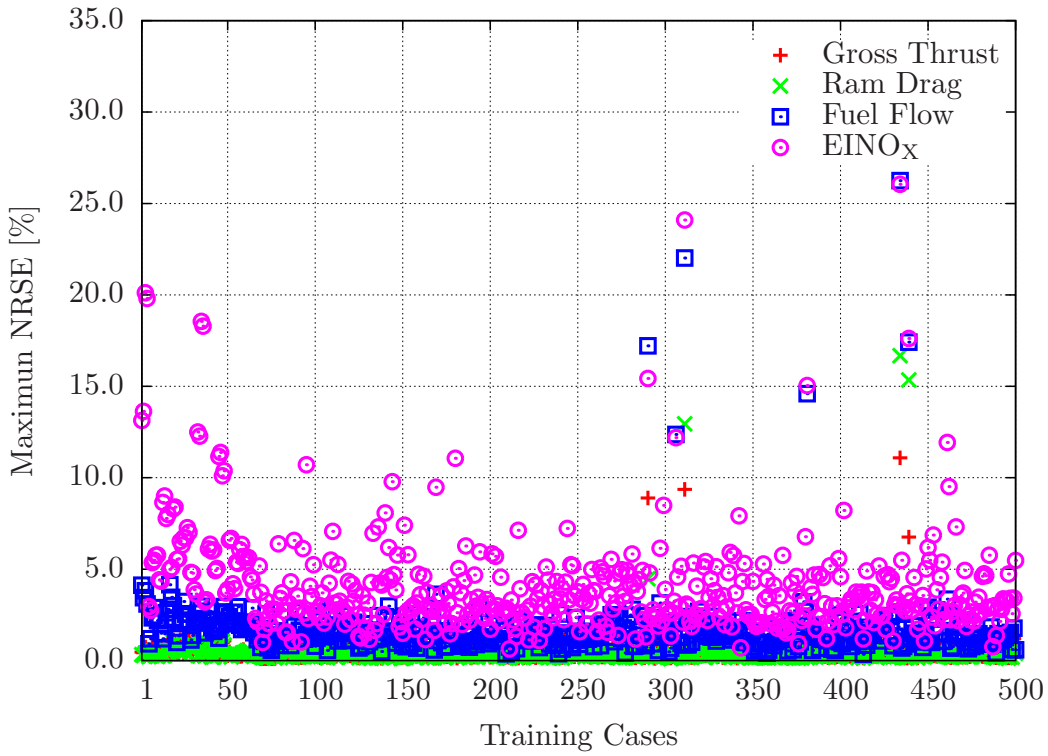
Table 11: NRMSE of the bases of engine deck responses between training and test data

	Gross thrust	Ram drag	Fuel flow	EINO_X
$\mathbf{v}_1$	$3.980957e{-}03$	$1.911276e{-}03$	$1.569202e{-}03$	$3.438867e{-}03$
$\mathbf{v}_2$	$1.335774e{-}02$	$1.261081e{-}02$	$1.608686e{-}02$	$1.797702e{-}02$
$\mathbf{v}_3$	$2.060794e{-}02$	$2.217635e{-}02$	$1.591716e{-}01$	$8.656342e{-}02$
$\mathbf{v}_4$	$2.969393e{-}02$	$3.934067e{-}02$	$8.206984e{-}02$	$4.129359e{-}02$

Regarding the validity of the weighting coefficient models, this research assesses the fitness of the coefficient prediction models with respect to the random test data in terms of an R^2 in Table 12. As shown in Table 12, the weighting coefficient models of gross thrust and ram drag are considerably reliable since their R^2 values are over 0.9 for all four of their coefficients. Unlike the weighting coefficient models of gross thrust and ram drag, not all coefficient models have high R^2 values in the cases of fuel flow and EINOx; for fuel flow, the R^2 values of the third and fourth coefficients are relatively low, and so is the fourth coefficient for EINOx. In Section 5.3.4.3, the effects of these low R^2-valued minor coefficients



(a) NRMSE



(b) Maximum NRSE

Figure 41: NRMSE and maximum NRSE for training data

will be discussed in connection with the prediction results of fuel flow and EINOx.

Table 12: R^2 of the coefficients of engine deck responses for test data

	Gross thrust	Ram drag	Fuel flow	$EINO_X$
α_1	$9.999939e{-}01$	$9.999968e{-}01$	$9.999952e{-}01$	$9.999953e{-}01$
α_2	$9.995829e{-}01$	$9.991308e{-}01$	$9.998484e{-}01$	$9.972685e{-}01$
α_3	$9.993541e{-}01$	$9.967606e{-}01$	$8.574791e{-}01$	$9.617943e{-}01$
α_4	$9.308675e{-}01$	$9.102620e{-}01$	$7.595319e{-}01$	$8.369770e{-}01$

5.3.4.3 Model Prediction Error Test

As with the previous prediction quality investigation in Section 5.3.4.1, the same examination process is repeated for 500 randomly-populated snapshots of engine deck responses not used for creating the reduced-order NPSS model. Unlike the earlier prediction tests with the training data, the assumed invariancy of the empirical bases of engine deck responses may not hold for the random test data unless the empirical bases are generated from training data that are representative of the entire design space. Since both the empirical bases and the weighting coefficient models are found to be considerably reliable in Section 5.3.4.2, the reduced-order model yields superb prediction results in Figures 42 and 43 despite its ignorance about the random test data. For instance, Figure 42 shows that R^2 values are at least over 0.9985. Figure 43 delineates that NRMSEs are overall less than 2% and increase to around 3.5% in Figure 43(a); likewise, maximum NRSEs are mostly less than 5% and skyrocket to around 35% in the 289^{th} test case in Figure 43(b). Again, despite the highest NRSE of EINOx, i.e., 35.16% in the 289^{th} test case, corresponding R^2 and NRMSE values are satisfactory such that 0.9999305 and 1.67%, respectively. Similar to the prediction test results for the training data, both gross thrust and ram drag show prediction accuracy superior to the other engine deck responses, namely fuel flow and EINOx. Although the minor coefficients of fuel flow and EINOx have relatively low R^2 values, the predicted fuel flow and EINOx are quite acceptable as shown in Figures 42 and 43. Overall, the prediction test results for the random test data in Figures 42 and 43 do not significantly differ from those for the training data in Figures 40 and 41. Therefore, comprehensive prediction investigations

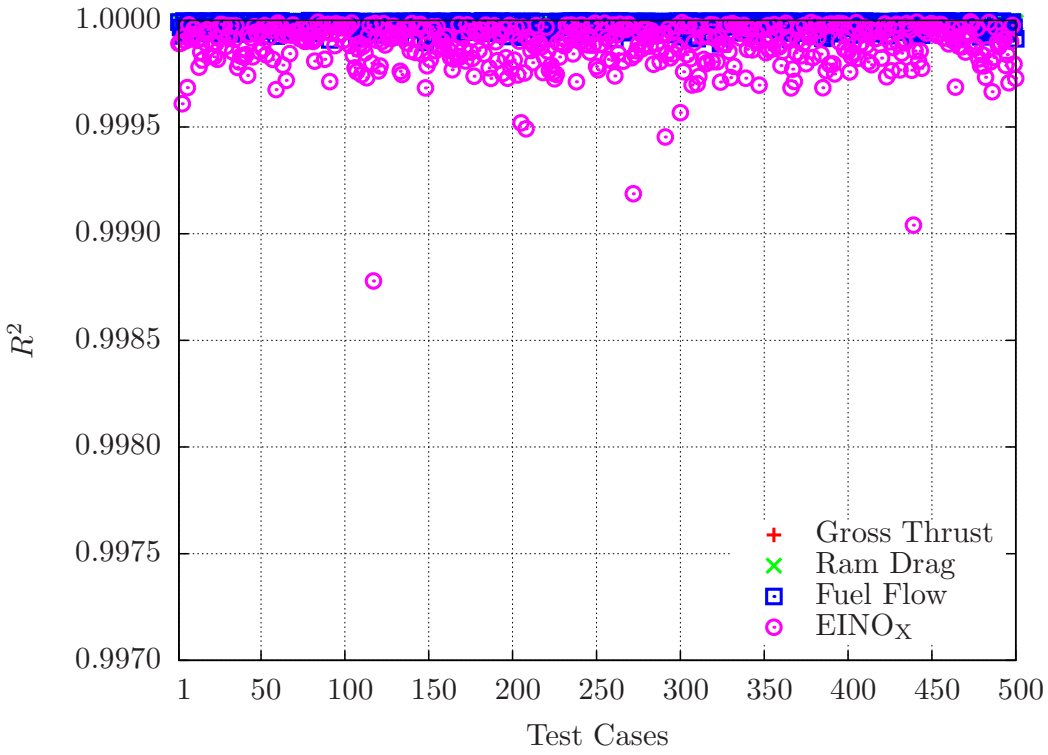
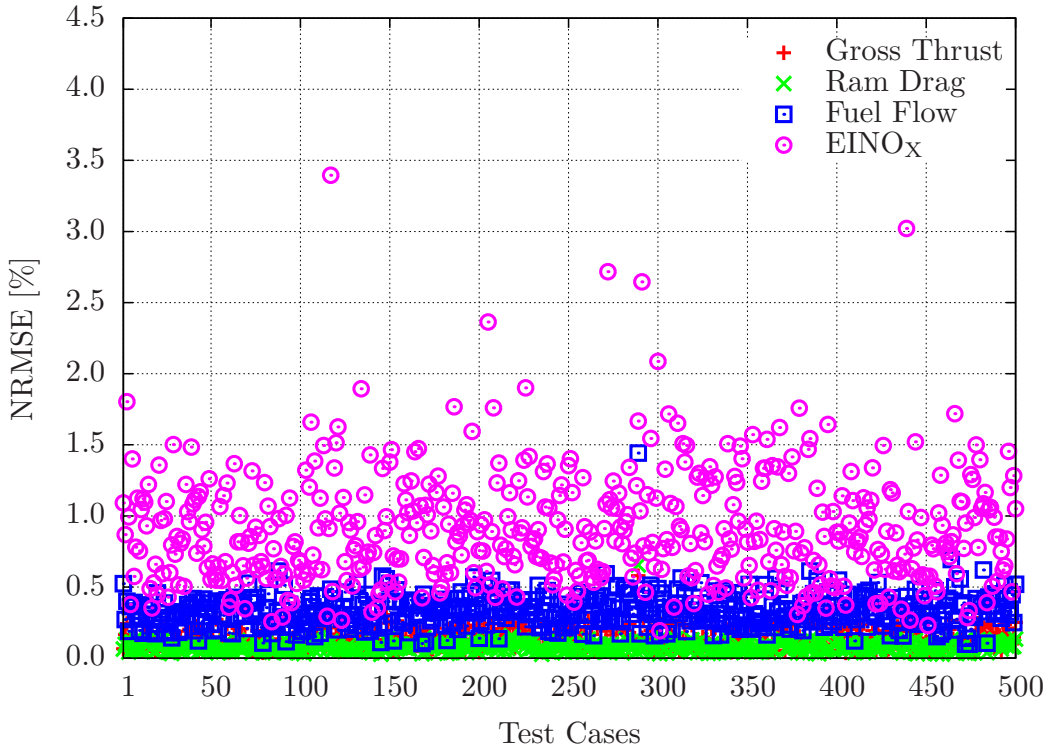


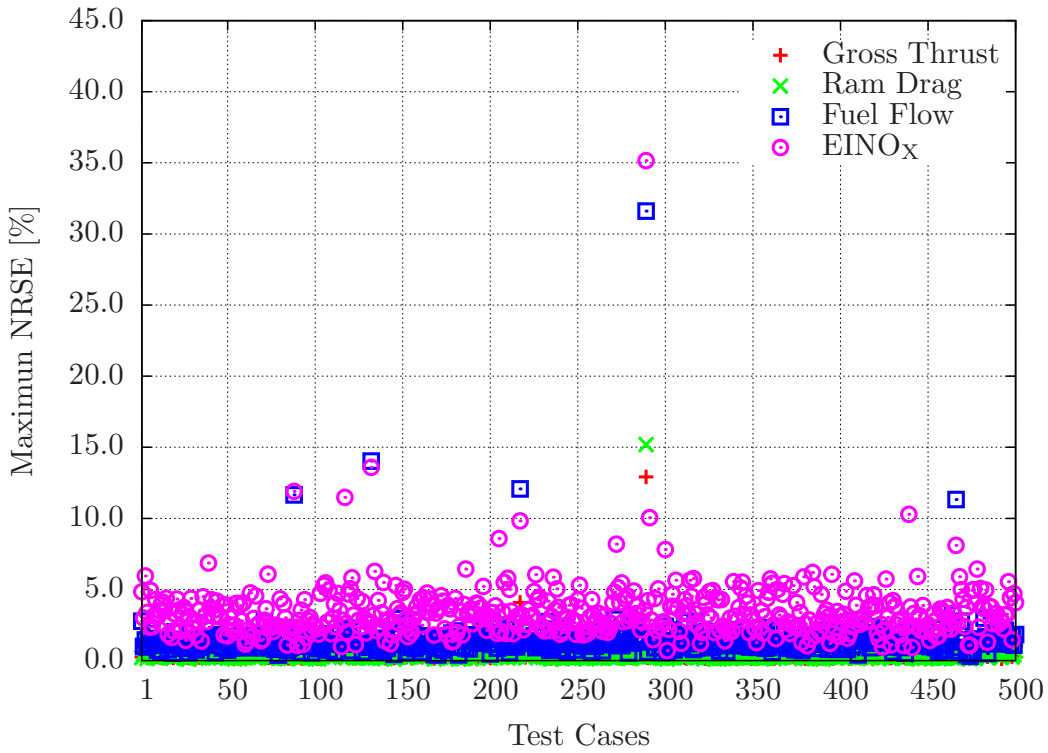
Figure 42: R^2 for random test data

with both the training and test data sets substantiate that the empirical bases as well as the weighting coefficient models are quite dependable, resulting in a credible reduced-order NPSS model for the given ranges of engine modeling parameters.

As shown in Figures 41(b) and 43(b), the maximum NRSEs in certain test cases are unacceptably high for all the engine deck responses. To further investigate this irregular behavior of engine deck responses, this research reanalyzes the 289^{th} test case, which showed the largest maximum NRSE in Figure 43(b). In Table 13, the previously obtained engine deck responses are compared to those newly-achieved around throttle value 22, at which the largest maximum NRSE occurred. Despite decreasing throttle values, NPSS originally resulted in engine deck responses exhibiting unusual increases at throttle value 22 due its convergence failure. After a Newton-Raphson solver within NPSS was adjusted so that it could generate converged solutions, the new NPSS analysis yielded engine deck responses showing concomitant decreases as the throttle value decreases. Numerically, these inconsistent NPSS results stem from the Newton-Raphson solver of NPSS for solving the equations of conservation of mass and energy in the process of determining engine performance in



(a) NRMSE



(b) Maximum NRSE

Figure 43: NRMSE and maximum NRSE for random test data

a given flight condition. Within NPSS, the solution of a flight condition was used as an initial guess for the subsequent flight condition. Should a flight condition fail to converge, then a solution for the maximum throttle at a given Mach number and altitude is substituted as a guess for the subsequent flight condition. Therefore, atypical engine performance data sometimes appear, especially at lower throttle settings. Thanks to the new converged NPSS results, the predicted engine deck responses obtained by the reduced-order NPSS model are now remarkably close to the newly-evaluated engine deck responses. Although deviant NPSS results do deteriorate the basis evaluation of engine deck responses, their effects on the empirical bases are minuscule since the test data set contains few outlying cases. After all, maximum NRSE values drastically drop, as delineated at the bottom of Table 13; for example, in the case of $EINO_X$, the maximum NRSE of 35.16145% is now 1.374225%.

Table 13: Comparison of NRSEs at Mach number = 0.40 and altitude = 20,000 ft.

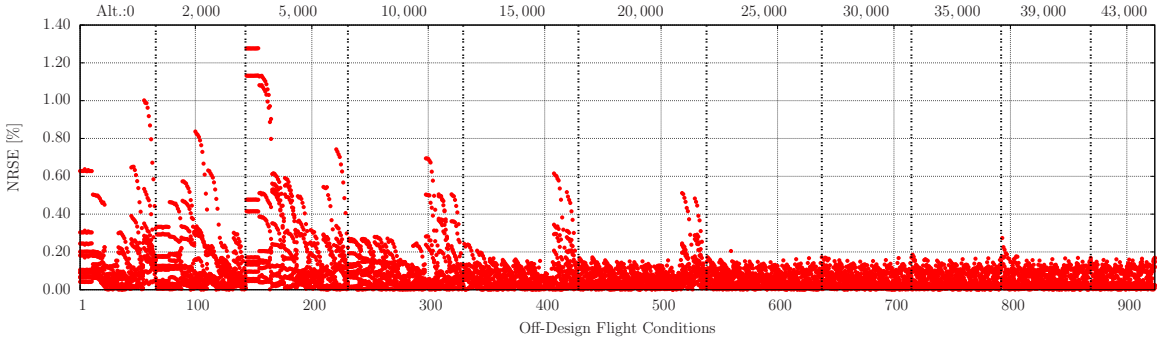
	Throttle Setting	Gross thrust [lbf]	Ram drag [lbf]	Fuel flow [lbm/hr]	$EINO_X$ [g/kg]
Old	24	15583.1	11590.4	2083.5	7.5379
	22	14618.6	12337.1	2292.1	8.5443
	21	12670.7	11242.3	1795.8	6.5093
New	24	15582.9	11590.4	2083.0	7.5638
	22	12750.4	10458.5	1557.5	5.4649
	21	11250.7	9809.7	1287.1	4.4680
Predicted	24	15563.4	11598.5	2092.4	7.6300
	22	12730.9	10463.5	1567.6	5.5400
	21	11231.5	9811.6	1300.4	4.5400
Old NRSE		12.91273%	15.18704%	31.61031%	35.16145%
New NRSE		0.152936%	0.047808%	0.648475%	1.374225%

As the final step of the prediction quality investigation, prediction errors at every 50^{th} snapshot are compiled in terms of an NRSE and plotted for all off-design flight conditions

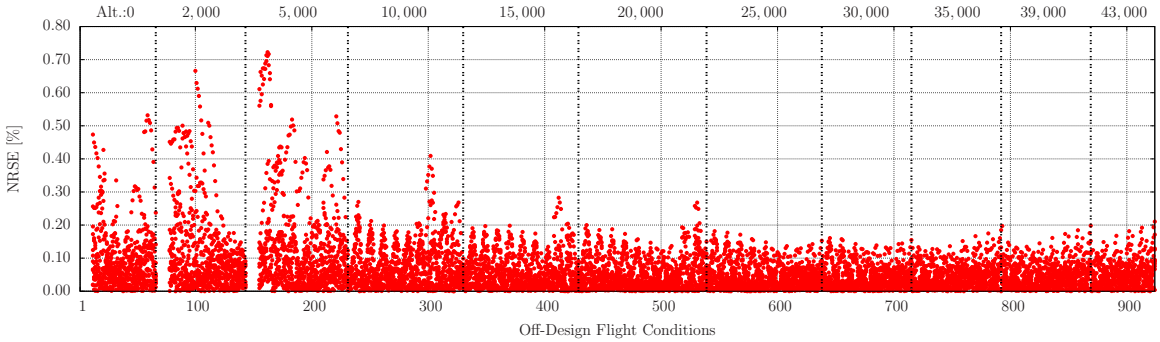
in Figure 44. For easier comprehension of the distributed prediction errors, Figures 44(a)–44(d) are segmented into 11 sections representing different altitudes in which a Mach number gradually increases as listed in Table 8. As a result, the decomposed figures in Figure 44 are conducive to one's anticipating the magnitude of relative prediction errors associated with certain flight conditions. For example, except EINOx, although the accumulated NRSEs of other engine deck responses are mostly low across all flight conditions, they start to settle down after 20,000 ft. By contrast, the NRSEs of EINOx are relatively larger than those of other engine deck responses, and they tend to increase after 20,000 ft. With respect to changes in the throttle setting, more prediction errors are observed at both maximum and minimum throttle settings rather than in-between throttle settings. Therefore, from the perspective of aircraft mission analysis, the reduced-order NPSS model is expected to be more precise in the cruise mode than in the climb and descend modes, both of which necessitate extreme throttle settings at low altitudes. Moreover, in connection with an environmental aspect, since EINOx is particularly interested around terminal areas whose altitudes are below 3,000 ft., one can expect approximately 4% of errors at most for predicted EINOx by the reduced-order NPSS model.

5.4 Summary

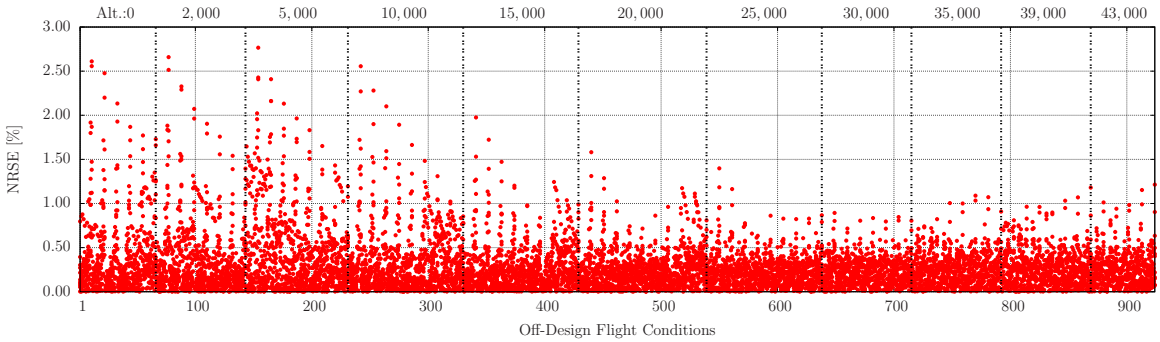
In order to facilitate the use of high-fidelity propulsion system simulations in aircraft design, this research formulated a reduced-order model for NPSS based on a POD-based ROM approach with the help of the EM-PCA and neural networks. Because of inevitable data absence observed in the results of NPSS for failed analyses, the EM-PCA is indispensable to achieving POD bases from NPSS engine deck responses. In addition, given the large number of engine modeling parameters in engine design, neural network models are effective at evaluating weighting coefficients in accordance with changes in the engine modeling parameters. As a demonstration, this research created a reduced-order NPSS model for a two-spool, separate flow turbofan in order to relate six engine modeling parameters with four engine deck responses whose dimensionality was 924. For the construction of the reduced-order model, a total of 500 sample data were populated as training data through



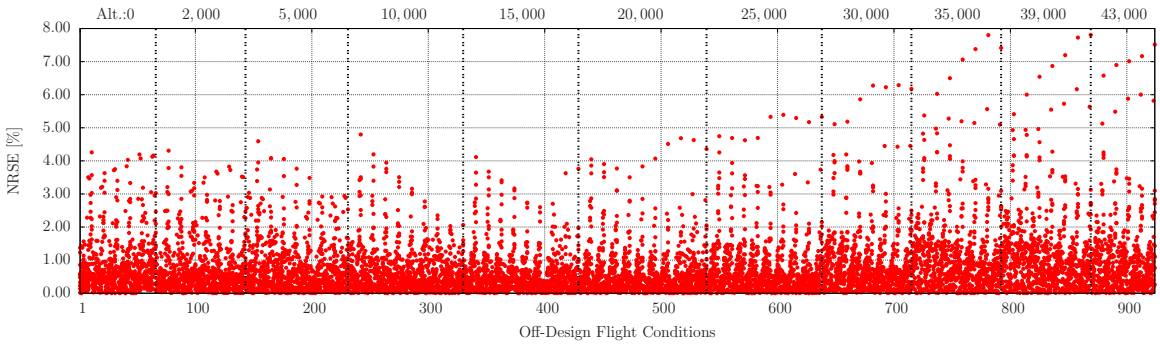
(a) Gross thrust



(b) Ram drag



(c) Fuel flow



(d) $\mathrm{EINO_X}$

Figure 44: NRSEs compiled at every 50^{th} snapshot

a maximum entropy design augmented with corner points, and the same number of data were randomly generated as test data.

By virtue of the EM-PCA and single-layer feed-forward networks, the reduced-order NPSS model built upon the first four dominant modes showed considerably low model-fit-errors in terms of an NRMSE and an R^2. Moreover, for the given random test data, the reduced-order model generated engine deck responses as accurate and reliable as they were for the training data. Although the reduced-order NPSS model yielded substantially dependable engine deck responses, the prediction qualities of gross thrust and ram drag were relatively higher than those of fuel flow and EINOx due to their superior weighting coefficient models. Due to the accurate prediction results of the reduced-order NPSS model at low computational cost, propagated errors to an aircraft system level would be insignificant and acceptable, encouraging one to capitalize on a reduced-order NPSS model for computationally intensive design studies.

Since the reduced-order NPSS model was prone to prediction errors mostly at low altitudes, the following two research tasks are recommended for future work. First, a domain decomposition approach could result in a locally refined reduced-order model that separately accounts for the different degrees of variation in engine deck responses with respect to altitude. Second, a weighted least-squares method could provide biased weighting coefficients such that they reduce more prediction errors at low altitudes than those at high altitudes. Furthermore, the synergy of these two proposed research ideas also could be beneficial to enhance the accuracy of the reduced-order NPSS model. Finally, with regard to the irregular increases of engine deck responses at low throttle settings, robust PPCA [8] could be useful for mitigating the outlier effects on empirical basis evaluations.

CHAPTER VI

APPLICATION II: EFFICIENT PIV DATA RESTORATION

6.1 Background

Among the various flow measurement methods, PIV is one of the most widely-used techniques such that it takes instantaneous images of a flowfield to calculate flow velocity. Although PIV provides high-fidelity information about the flowfield of interest, achieved velocity measurements through PIV usually necessitate post-processes for the validation of the raw data.[61] For instance, PIV recordings could be contaminated with dubious observations for the following reasons:[57,58] (i) poor image contrast for irregular illumination, (ii) low and inconsistent seeding density, and (iii) an ill-prepared experimental setup. In addition, as Venturi and Karniadakis[84] pointed out, an obstructed view in an experimental setting and adjacency to boundaries may produce deficient measurements. Even though these spurious observations can be corrected by local means or interpolations of neighboring points, this approach may yield physically improper estimates, specifically when unreliable observations reside in the region of high nonlinearity such as vorticities.

To deal with erroneous or occasionally unseen data, several researchers have investigated the utility of gappy POD, formulated by Everson and Sirovich.[13] Originally, gappy POD was devised to recover marred human facial images; however, it is also applicable to repairing PIV data. After all, both problems are identical in the sense of missing data estimation once impaired data are removed for restoration. For instance, Venturi and Karniadakis[84] and Gunes, Sirisup, and Karniadakis[16] examined gappy POD in comparison with other reconstruction methods such as local linear interpolation and local kriging, illustrating that gappy POD is superior in case of either a large snapshot ensemble or small gappiness. Furthermore, Murray and Ukeiley[58] and Murray and Seiner[57] applied gappy POD to PIV data acquired from subsonic cold jet experiments, demonstrating that gappy POD can improve deficient data so that they are accurate enough to the level of experimental uncertainty.

From a formulation aspect, gappy POD is the deterministic enhancement of POD for missing data estimation whereas PPCA is a probabilistic extension of POD, i.e., PCA. For a probability model of PCA, Tipping and Bishop[82] developed PPCA, which yields a Gaussian probability model. After utilizing the EM algorithm[10] for parameter estimation, Tipping and Bishop[82] derived an iterative algorithm referred to as the EM-PCA. Because the EM algorithm handles missing data by nature, the EM-PCA is also capable of approximating missing observations while deriving probability parameters. The EM-PCA has been mostly employed in the context of image processing and pattern recognition, e.g., speech recognition.[70] Later on, in other engineering fields, Lee, Rallabhandi, and Mavris[35] examined its feasibility for both basis extraction and missing data estimation with simulation data collected from CFD analysis.

Although both gappy POD and the EM-PCA depend on POD, they end up with different algorithms for their dissimilar formulations. To delve into their similarities and disparities, Lee and Mavris[32,33] proposed a unifying least-squares perspective and revealed that their rudimentary difference stems from their dissimilar bases and norms. Furthermore, Lee and Mavris[30] scrutinized the effects of the different bases and norms, unveiling that the norm is a predominant factor affecting missing data estimation rather than the basis. According to Lee and Mavris,[30,32,33] both the basis and the coefficient evaluation of the EM-PCA are simpler to compute than those of gappy POD. In addition, owing to the coefficient formulation of gappy POD, the performance of gappy POD is vulnerable to such a missing data type that has spared missing data. Therefore, the EM-PCA is more advantageous than gappy POD to restoring spurious PIV data scattered over an entire snapshot ensemble.

In summary, the goal of this chapter is to introduce the EM-PCA as an efficient alternative to gappy POD for PIV data restoration. Overall, the outline of this chapter is organized as follows. First, it delineates the process of PIV data generation and illustrates experimental settings for the test of acoustically-excited, bluff-body jet flow. Afterwards, it describes the deterministic and probabilistic POD-based approaches for missing data estimation, i.e., gappy POD and the EM-PCA, with a particular focus on their computational costs associated with their formulations. In the next section, it validates the results of

the EM-PCA with those of gappy POD in terms of eigenspectra, flow velocity modes, and restored PIV data. In the last and most important section, it illustrates the computational performance of both methods through several numerical experiments.

6.2 Experimental Data Generation

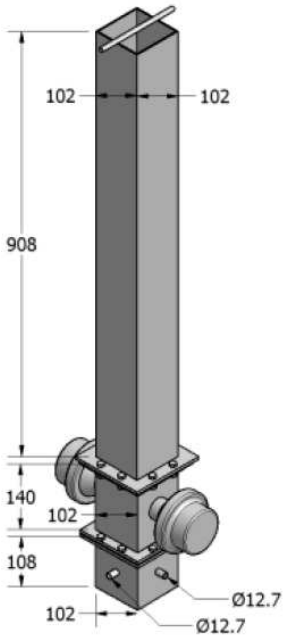


Figure 45: The experimental apparatus for the test of a bluff-body reacting jet-flow with acoustic excitation; dimensions in millimeters[73]

The details of an experimental apparatus and the procedure of PIV measurement are given in the work of Shanbhogue et al.;[72] however, they are summarized herein for completeness. Shanbhogue et al.[72] conducted bluff-body, stabilized reacting jet-flow experiments with harmonic acoustic excitation to investigate combustion instabilities. As depicted in Figure 45, the atmospheric pressure burner has a square-shaped exit with a bluff-body installed at the top of the channel. At the bottom of the channel, a cyclone seeder disperses Al_2O_3 flakes whose sizes range from 0.9 to 2.2 μm among the mixture of methane and air. The seeded mixture passes through a honeycomb-grid straightening section, and they are acoustically excited by speakers above the flow straightener. Two 100 W Walsh PA loudspeakers are located to exert sinusoidal acoustic force produced by an Agilent 33120A-15 MHz function generator along with a 100 W RadioShack MPA-101 amplifier. Finally, the

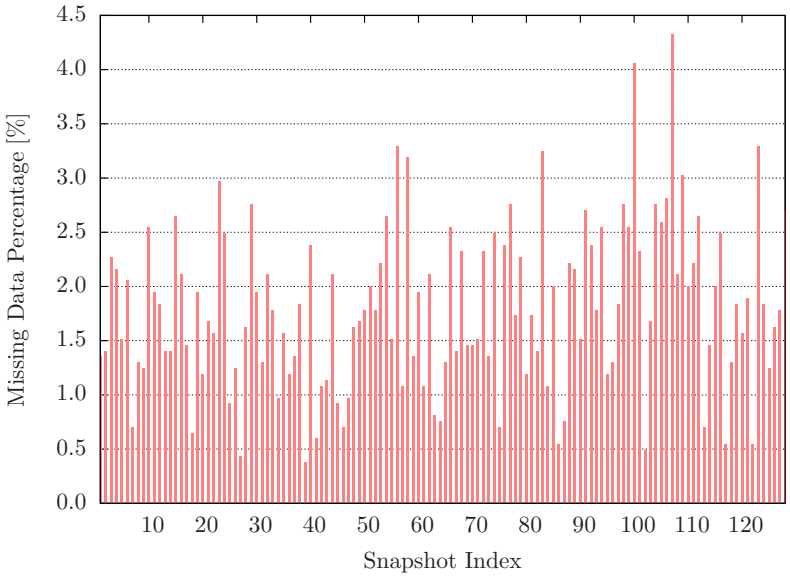
acoustically-excited mixture burns right after a bluff body mounted at the exit of the channel. For a flame stability study, both triangular and circular cross-sectioned bluff bodies were utilized, but this research uses PIV measurements with the triangular bluff body only in the remaining discussion.

To gauge flow velocity, the PIV apparatus is composed of a dual head Nd:YAG laser and a digital camera. The Nd:YAG laser emits light with a wavelength of 532 nm, a peak power of 120 mJ/pulse, and a pulse duration of 30 μs. The camera is equipped with a 1600-by-1200 pixel CCD image sensor whose pixel size is 7.4 μm as well as a Nikon F-mount 55 mm micro-lens with an aperture of f/5.6. For the conversion of a laser light into a thin laser sheet, two cylindrical lenses whose focal lengths are 150 and 1000 mm were employed. The two cylindrical lenses reduce the laser beam thickness to 5 mm, causing the light beam to diverge to the height of 40 mm. From the camera, an imaging plane was set at a distance of 304.8 mm, and a view field was sized at 38.1 mm by 28.575 mm with a conversion rate of 41.99 pixels/mm.

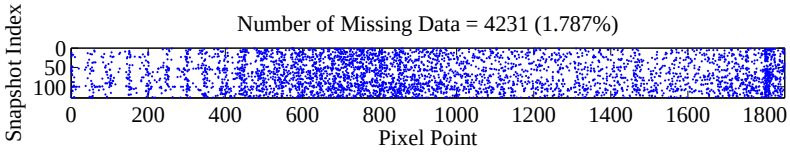
For a combustion instability study, reacting jet-flow experiments were carried out at various conditions of sinusoidal acoustic excitation: (i) a phase angle from 0° to 240°, (ii) an amplitude from 1 V to 5 V, and (iii) a frequency from 300 Hz to 600 Hz at the inverse of 150Hz. At each phase angle, a total of 128 snapshots of images were collected and ensemble-averaged, resulting in approximately 2% of the velocity uncertainty. From the multiple PIV data sets at diverse experimental settings, Lee and Mavris[31] utilized snapshots tested in the following conditions: approaching velocity 3.62 m/s, inlet temperature 298 K, phase angle 225°, amplitude 1 V, and frequency 450 Hz.

To process compiled images, Lee and Mavris[31] used DaVis software by LaVision, partitioning the view field into a 32-by-32 pixel grid with a spatial resolution of 0.762 mm. With the help of DaVis, Lee and Mavris[31] were able to eliminate 1.787% of spurious PIV data, generating a 1850-by-128 incomplete data set for each u and v velocity component. As an illustration, Figure 46(a) delineates the variations of missing data rates across snapshots, showing that the maximum missing rate is 4.32432%. In addition, Figure 46(b) depicts the locations of missing data scattered over an entire snapshot ensemble. In Figure 46(b), an

abscissa represents a measurement point treated as one dimensional, and an ordinate is a snapshot index. Note that missing data tend to cluster at certain locations, indicating that some measurement points are prone to produce misleading observations.



(a) Missing data percentage



(b) Missing data distribution

Figure 46: Missing PIV measurements

6.3 Validation with Restored PIV Data

Before a validation study, this section determines the adequate number of modes q for each velocity snapshot ensemble, comparing restored eigenspectra at different q values. Once q is set, the results of EM-PCA implementations are compared with those of gappy POD implementations in terms of eigenspectra, flow velocity modes, and restored velocity fields.

6.3.1 Algorithm Implementations

In order to implement both algorithms of gappy POD and the EM-PCA, this research constructed them with the following options: (i) allowance for a sample mean change, (ii) a $\mathbf{W}$ initialization manner, and (iii) a $\mathbf{V}_q$ evaluation scheme. Among these three options,

Table 14: Notations of various implementations for gappy POD and the EM-PCA

Name	$\boldsymbol{\mu}$	$\mathbf{W}$	$\mathbf{V}_q$
EM-PCA	$\boldsymbol{\mu}$ inv./$\boldsymbol{\mu}$ var.	`rand`/$\mathbf{V}_e$	
GPOD	$\boldsymbol{\mu}$ inv./$\boldsymbol{\mu}$ var.		SVD/Lanczos

the first is applied to both algorithms, but the second and the third are related only to the EM-PCA and gappy POD, respectively. In the first place, depending on whether a sample mean is computed at an iteration, both gappy POD and EM-PCA implementations have two versions. For example, implementations whose sample means are invariant during iterations are denoted by "$\boldsymbol{\mu}$ inv.," and the others whose sample means are variant are indicated by "$\boldsymbol{\mu}$ var." in their name.

Second, EM-PCA implementations branch out into two variants based on how $\mathbf{W}$ is initialized; a random initialization is represented by "`rand`," and an informed initialization with a POD basis is specified by "$\mathbf{V}_e$." This informed initialization for $\mathbf{W}$ takes advantage of an estimated POD basis that gappy POD uses to initiate its iterations. Third, gappy POD implementations have a derivative that exploits the Lanczos algorithm to expedite a POD process; implementations employing the Lanczos algorithm are entitled by appending "Lanczos" in their names; otherwise, no additional notation is used in their names. For the Lanczos algorithm, this research utilized the MATLAB function `eigs` with proper options; internally, `eigs` relies on the Fortran Library ARPACK.[38] Note that gappy POD implementations with the Lanczos algorithm are mainly tested in the context of a performance investigation. Overall, a total of four implementations are realized for each gappy POD and EM-PCA algorithm through the combination of the options listed in Table 14.

Regarding convergence determination, a RMSR of an estimated snapshot $\widetilde{\mathbf{y}}_j$ defined as

$$\mathrm{RMSR}^{(k)} = \sqrt{\frac{1}{dN}\sum_{j=1}^{N}\left\|\widetilde{\mathbf{y}}_j^{(k)} - \widetilde{\mathbf{y}}_j^{(k-1)}\right\|_{L^2}^2}$$

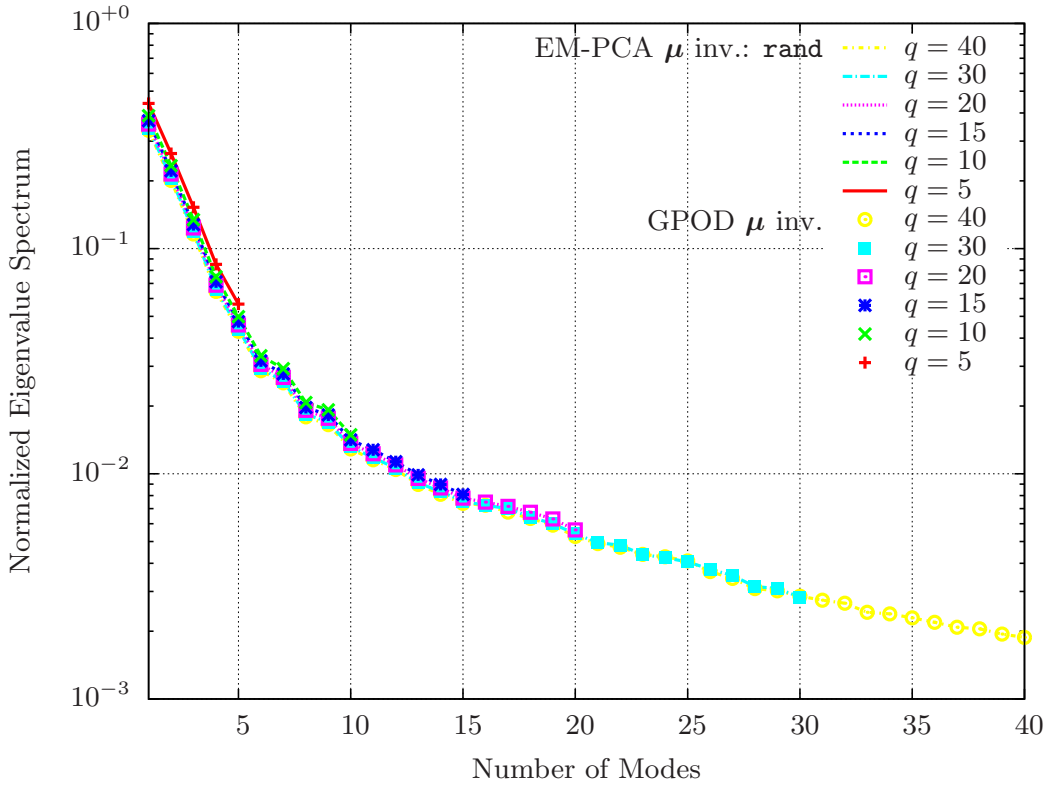
is monitored after it is normalized with respect to the first RMSR. In addition, an iteration number is inspected to prevent excessive iterations. All in all, convergence thresholds for a normalized RMSR and an iteration number are set to 10^{-6} and 10^4, respectively.

For numerical performance tests, all implementations were executed in a MATLAB R2007b environment on a PC equipped with an Intel Pentium dual-core 2.8 GHZ processor and 1 GB memory. The computational times were measured with the MATLAB `tic` and `toc` functions. Note that the EM-PCA implementations using random initialization were successively run 100 times for the minimal random effect on computational time assessment.

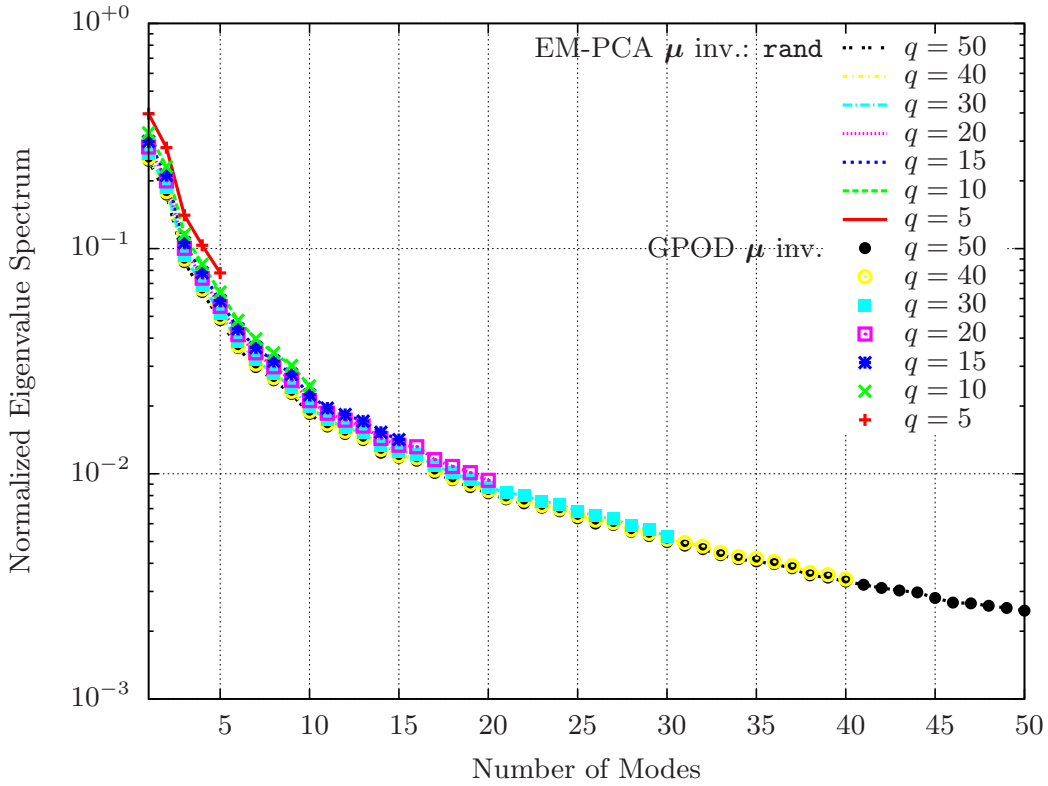
6.3.2 Selection of the Optimal Number of Modes

Venturi and Karniadakis[84] suggested that the optimal number of modes can be found when the eigenspectra of the restored data show no more changes with the q increment. Thus, Lee and Mavris[31] scrutinizes the eigenspectra of the u and v velocity snapshot ensembles as q increases from $q = 5$ to $q = 40$ at intervals of 5 or 10. In Figure 47, the first two Figs. 47(a) and 47(b) delineate the results of the $\boldsymbol{\mu}$ invariant methods, and similarly, the next two Figs. 47(c) and 47(d) depict those of the $\boldsymbol{\mu}$ variant methods. Moreover, Figure 47 arranges the eigenspectra of the restored u and v snapshot ensembles on the top and on the bottom, respectively. Note that the randomly initialized EM-PCA implementation is employed for eigenspectrum examination since the other EM-PCA implementation, initialized with $\mathbf{V}_e$, produces identical results.

Regardless of the snapshot ensembles, Figure 47 conveys that the eigenspectra obtained with the EM-PCA implementations, denoted in lines, are consistent with those achieved with the gappy POD implementations, represented in dots. In Figure 47, each gappy POD and EM-PCA implementation yields eigenspectra that are a little incongruent at different q values since its normalizing sum of eigenvalues keeps increasing with q; hence, the dwindling eigenspectra with q exhibit convergence at a large q value. For example, the eigenspectrum of the u snapshot ensemble almost settles down after $q = 30$ in Figs. 47(a) and 47(c), as does the eigenspectrum of the v snapshot ensemble after $q = 40$ in Figs. 47(b) and 47(d). Therefore, for a further validation study, this paper decides to set $q = 40$ and $q = 50$ for the u and v snapshot ensembles, respectively, in a conservative sense.

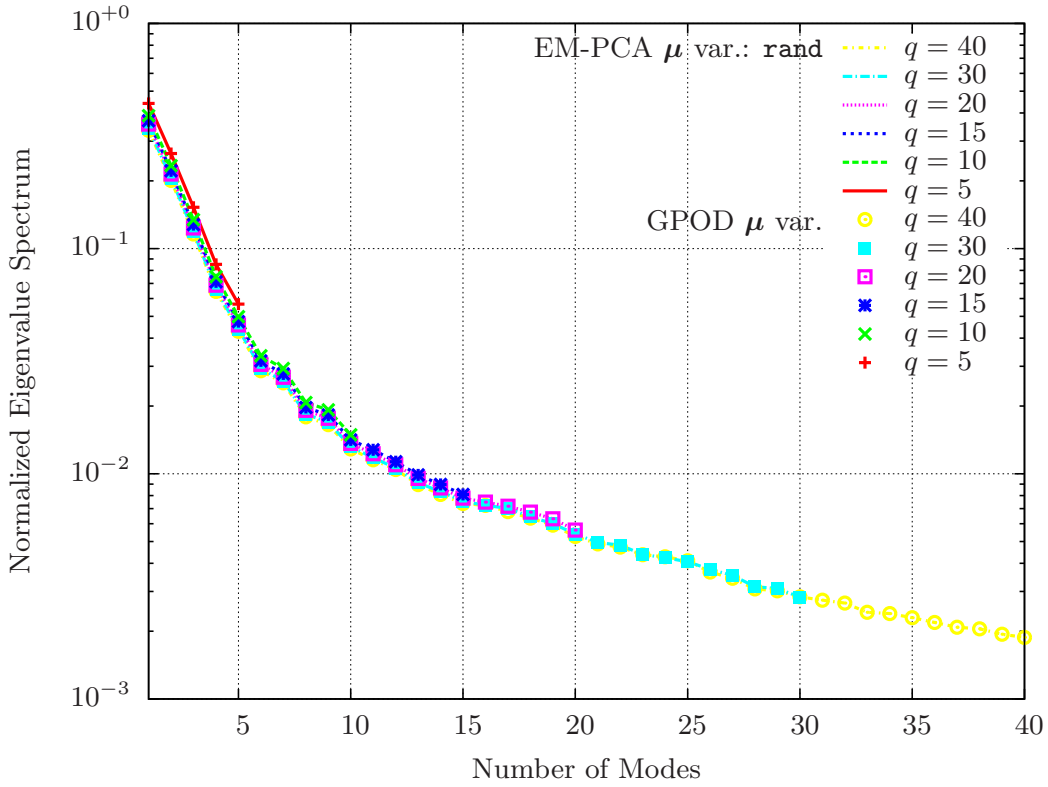


(a) u snapshot ensemble with the $\boldsymbol{\mu}$ invariant methods

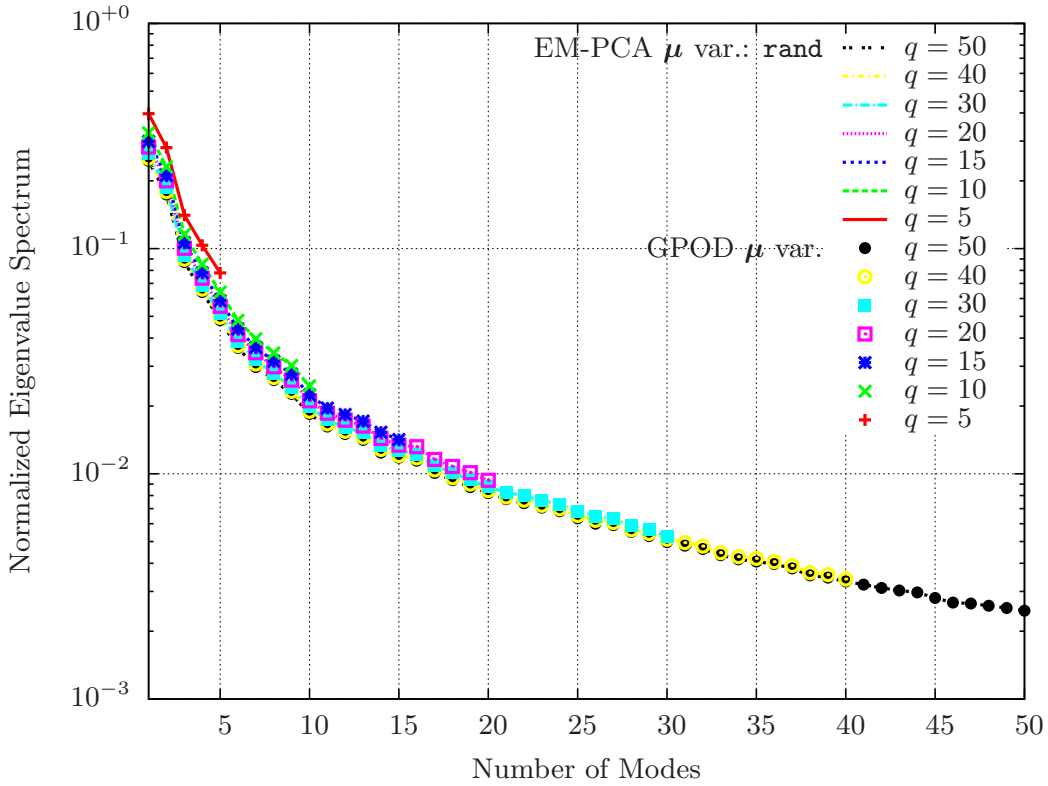


(b) v snapshot ensemble with the $\boldsymbol{\mu}$ invariant methods

Figure 47: Eigenspectra of restored u and v velocity components with q changes

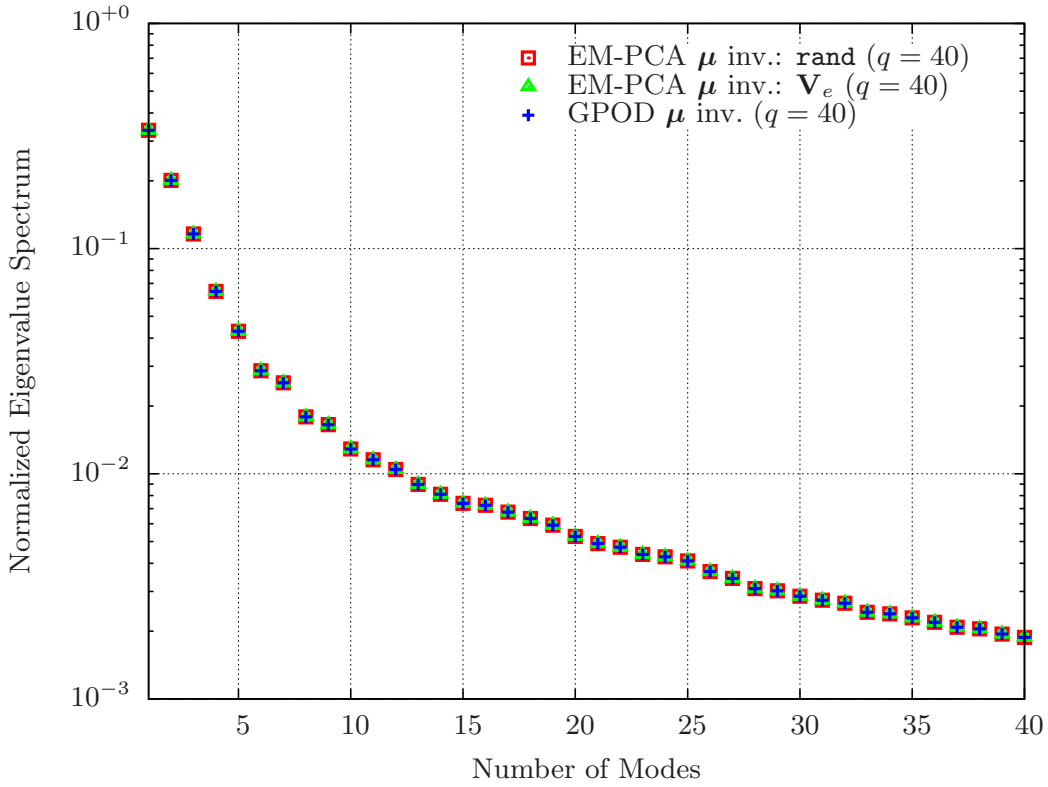


(c) u snapshot ensemble with the $\boldsymbol{\mu}$ variant methods

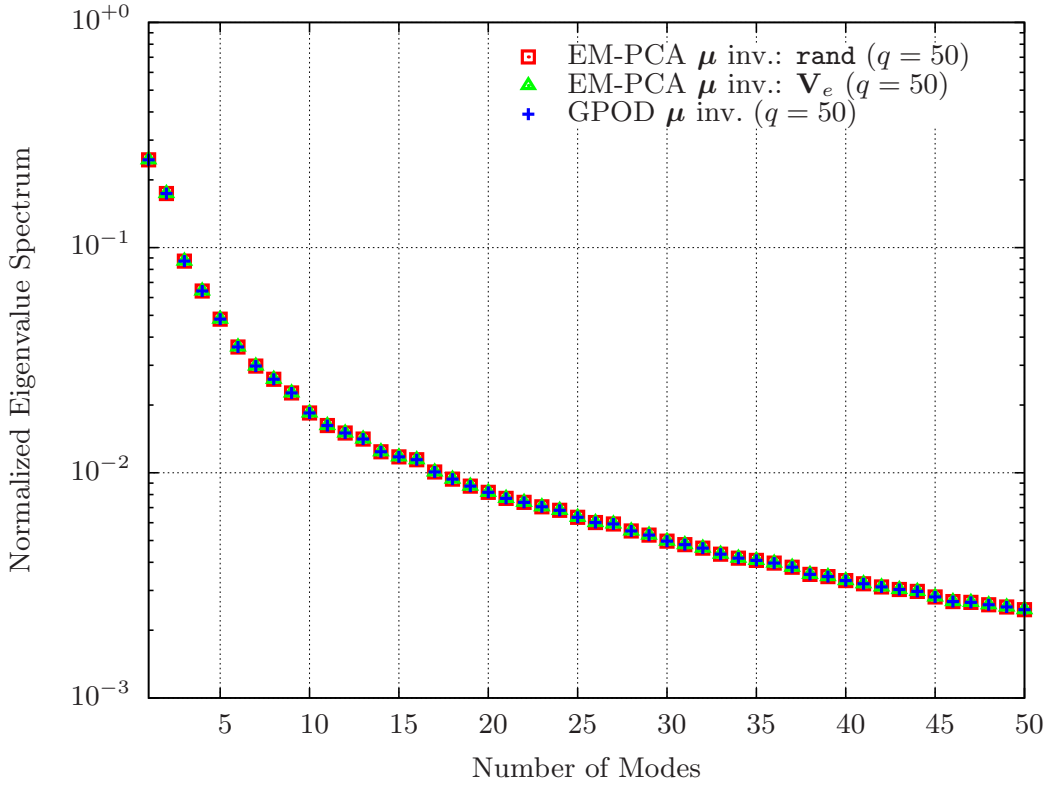


(d) v snapshot ensemble with the $\boldsymbol{\mu}$ variant methods

Figure 47: Eigenspectra of restored u and v velocity components with q changes

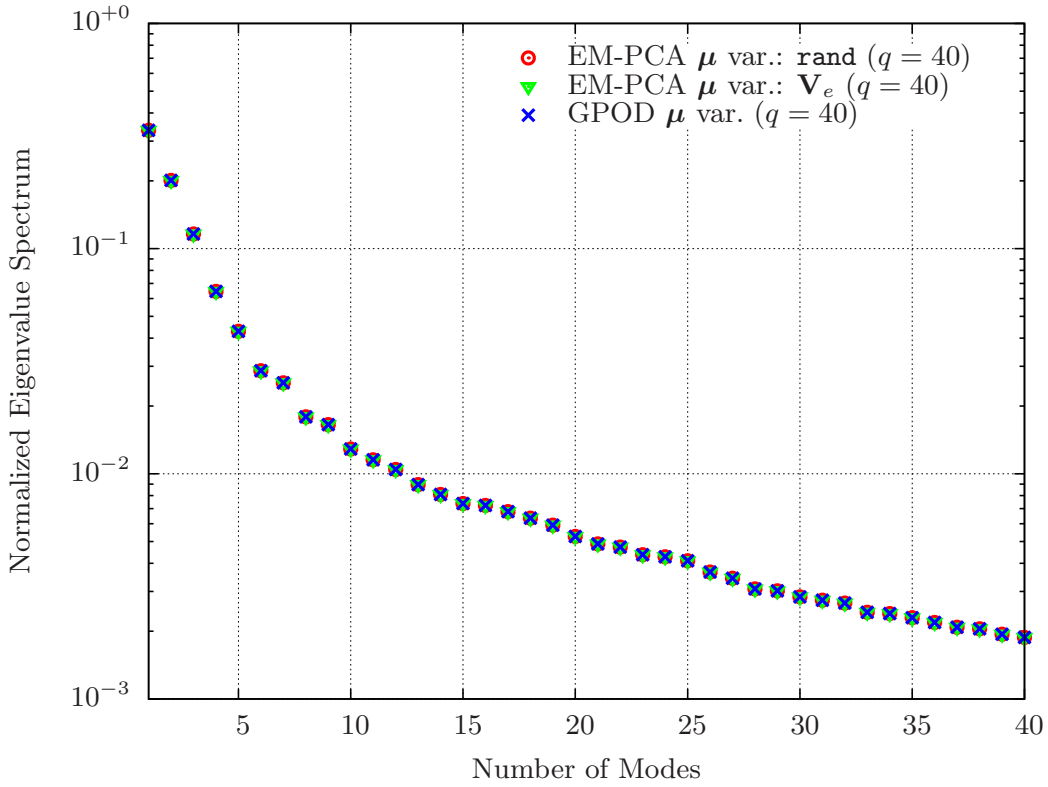


(a) u snapshot ensemble

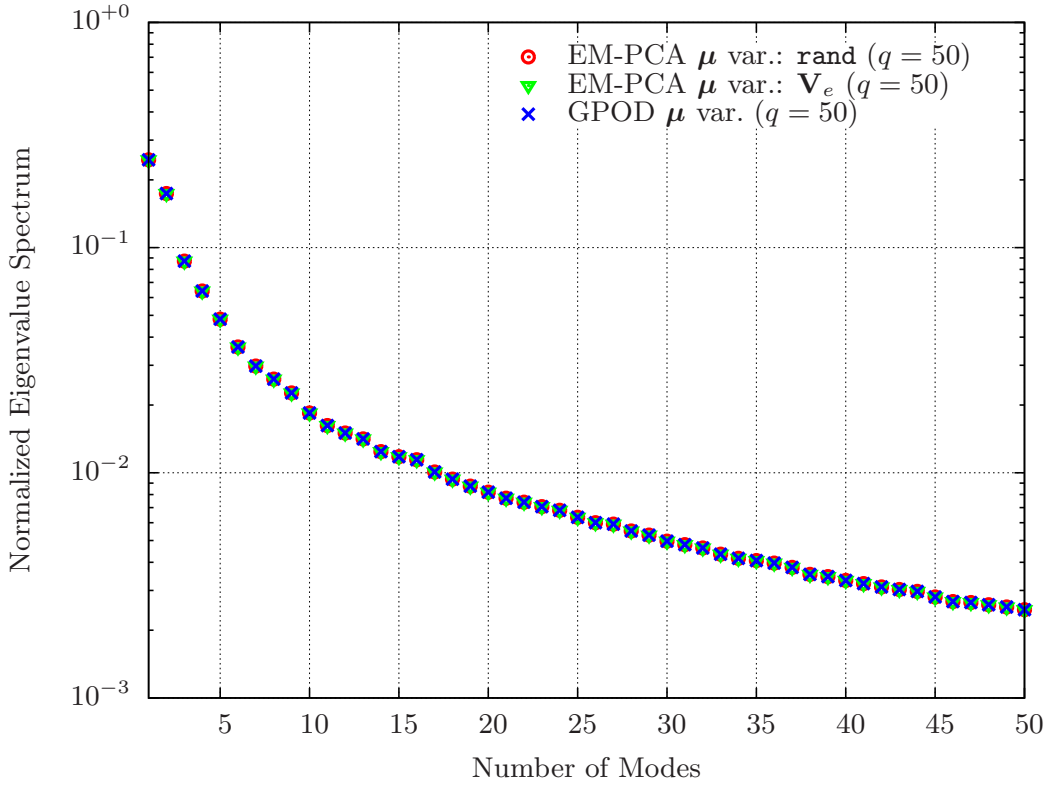


(b) v snapshot ensemble

Figure 48: Eigenspectra of restored u and v velocity components

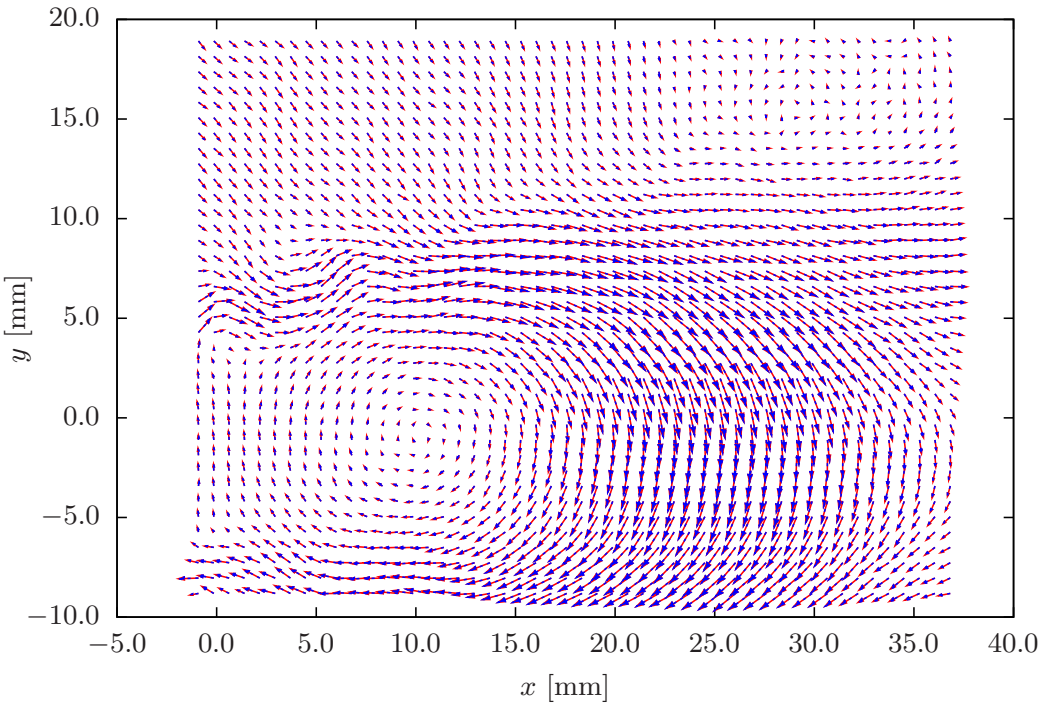


(c) u snapshot ensemble

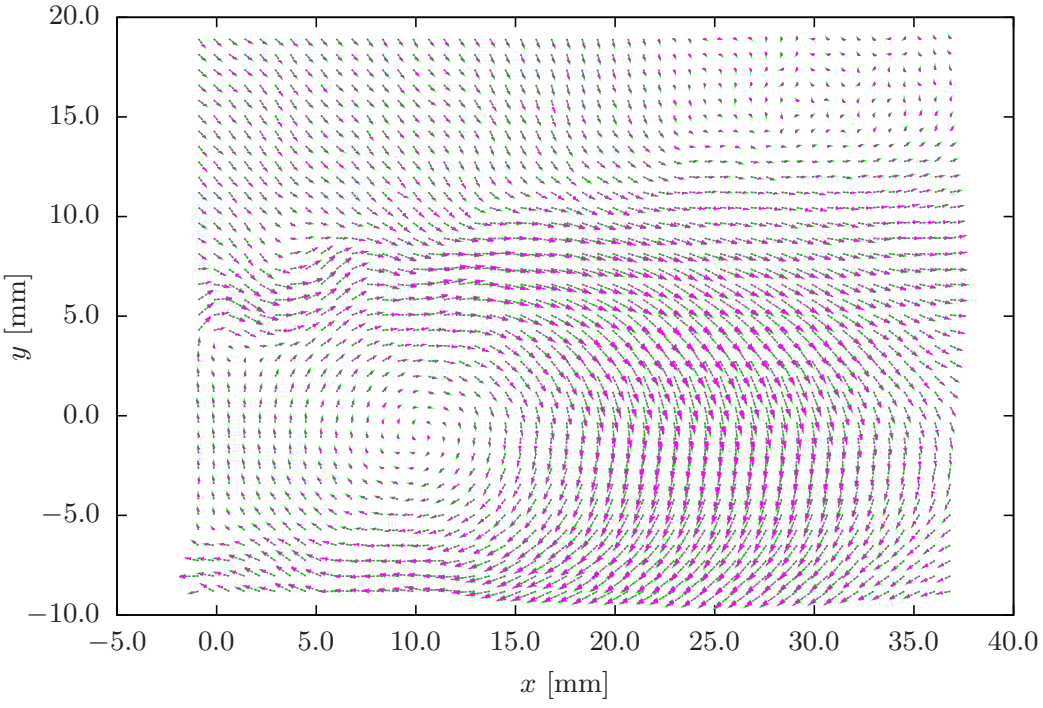


(d) v snapshot ensemble

Figure 48: Eigenspectra of restored u and v velocity components

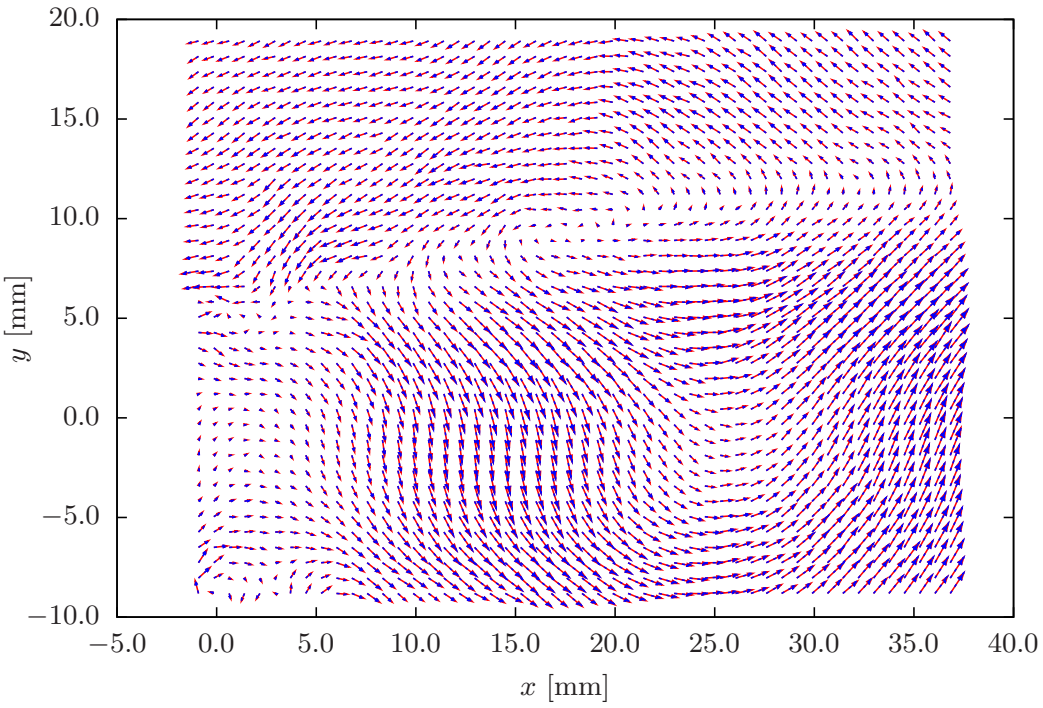


(a) $\boldsymbol{\mu}$ invariant methods

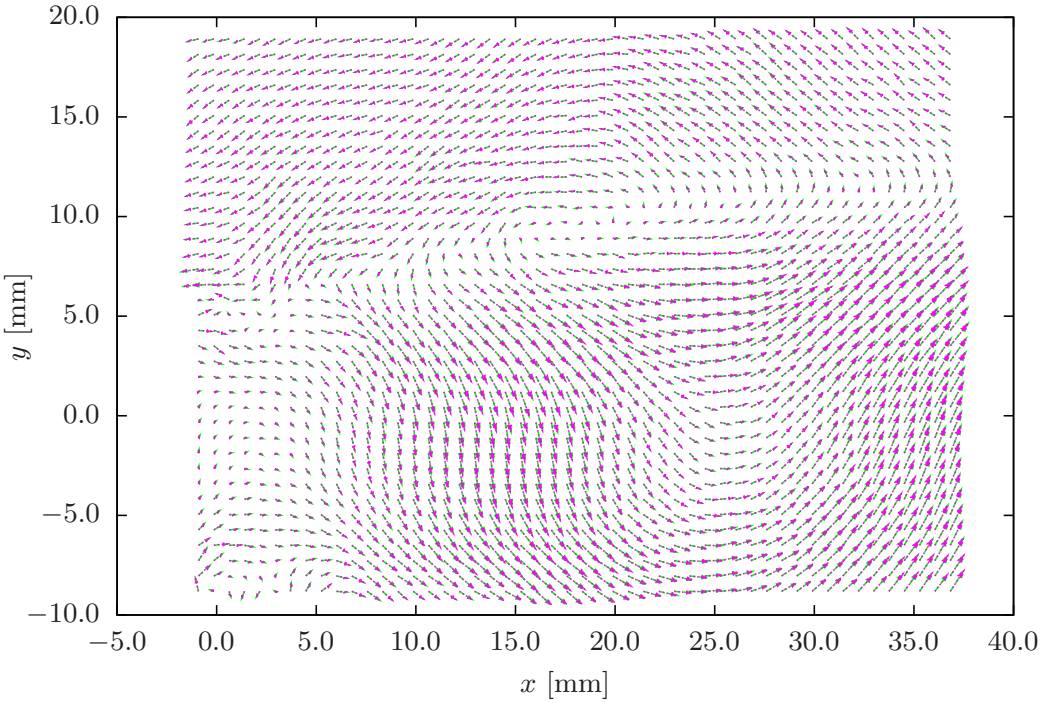


(b) $\boldsymbol{\mu}$ variant methods

Figure 49: 1st flow velocity modes of restored u and v velocity components: $u(q = 40)$, $v(q = 50)$

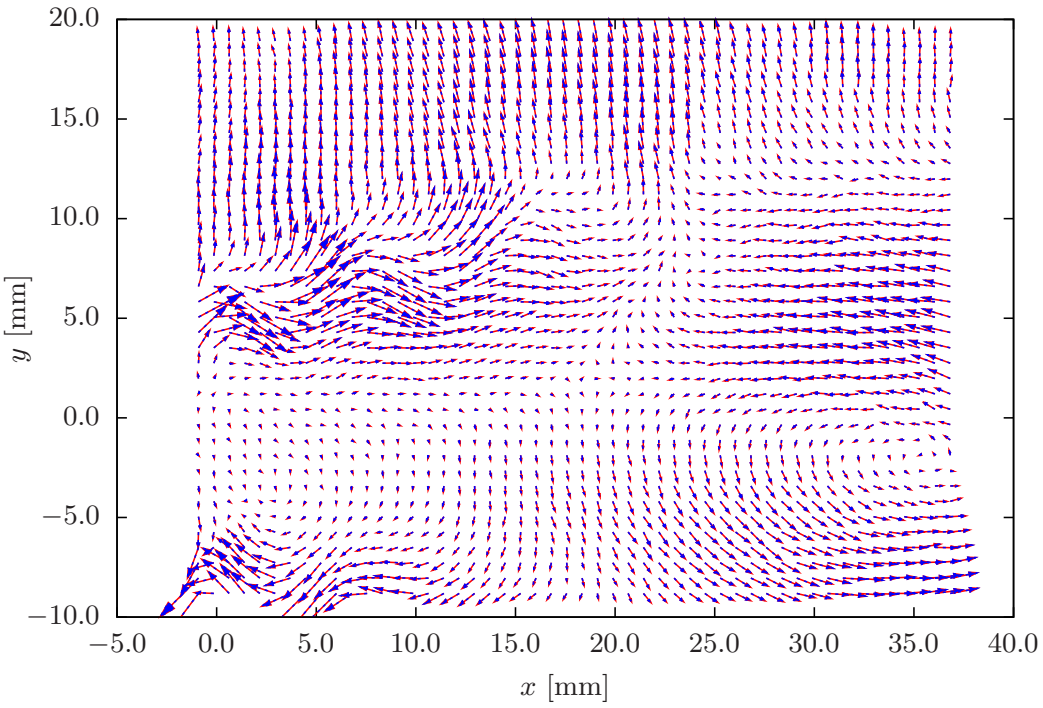


(a) $\boldsymbol{\mu}$ invariant methods

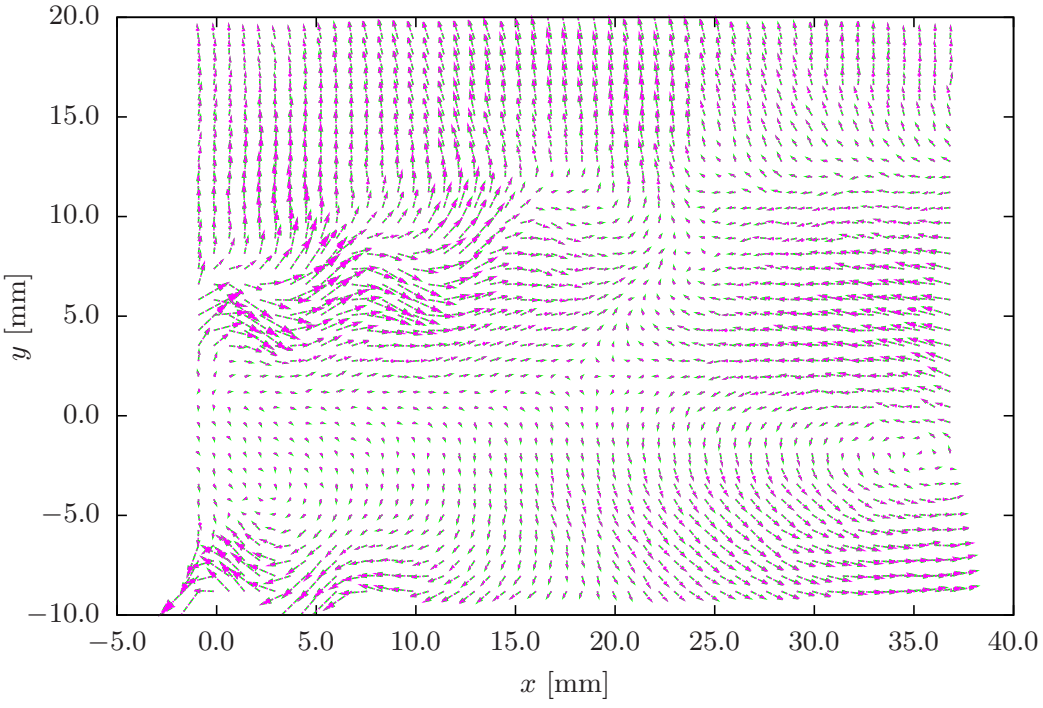


(b) $\boldsymbol{\mu}$ variant methods

Figure 50: 2nd flow velocity modes of restored u and v velocity components: $u(q = 40)$, $v(q = 50)$

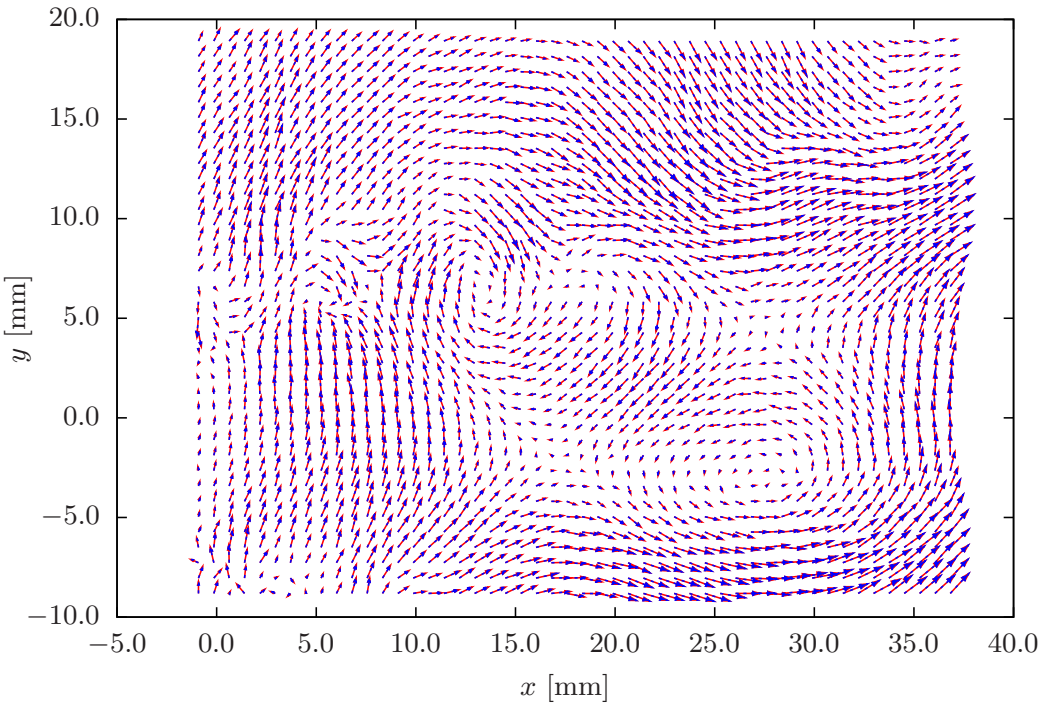


(a) $\boldsymbol{\mu}$ invariant methods

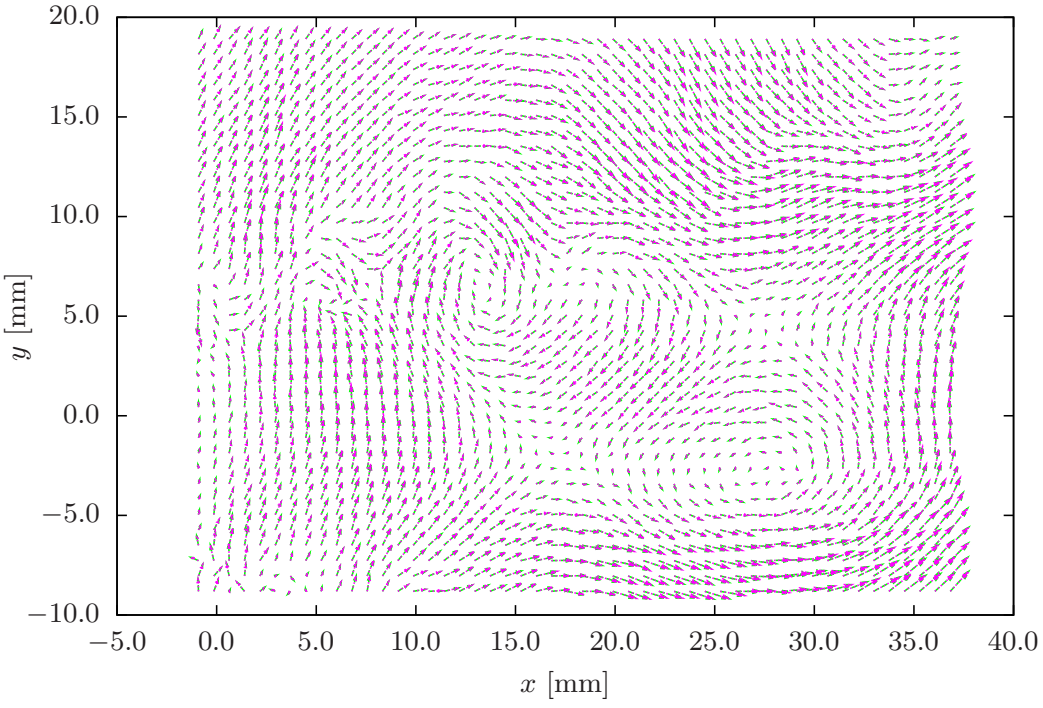


(b) $\boldsymbol{\mu}$ variant methods

Figure 51: 3rd flow velocity modes of restored u and v velocity components: $u(q = 40)$, $v(q = 50)$

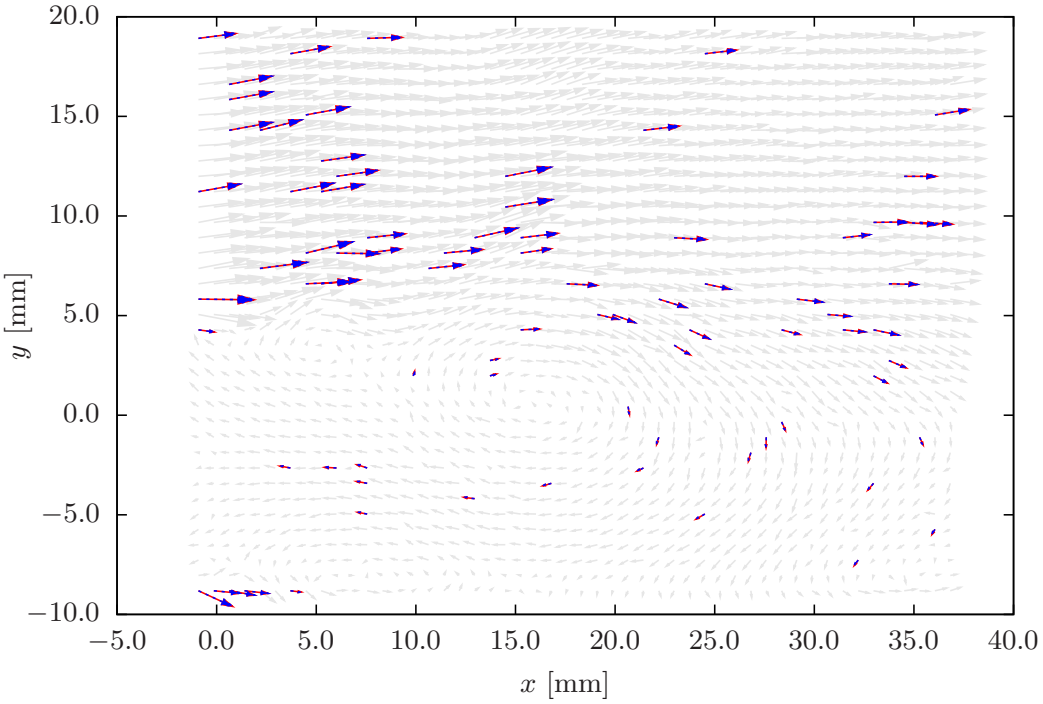


(a) $\boldsymbol{\mu}$ invariant methods

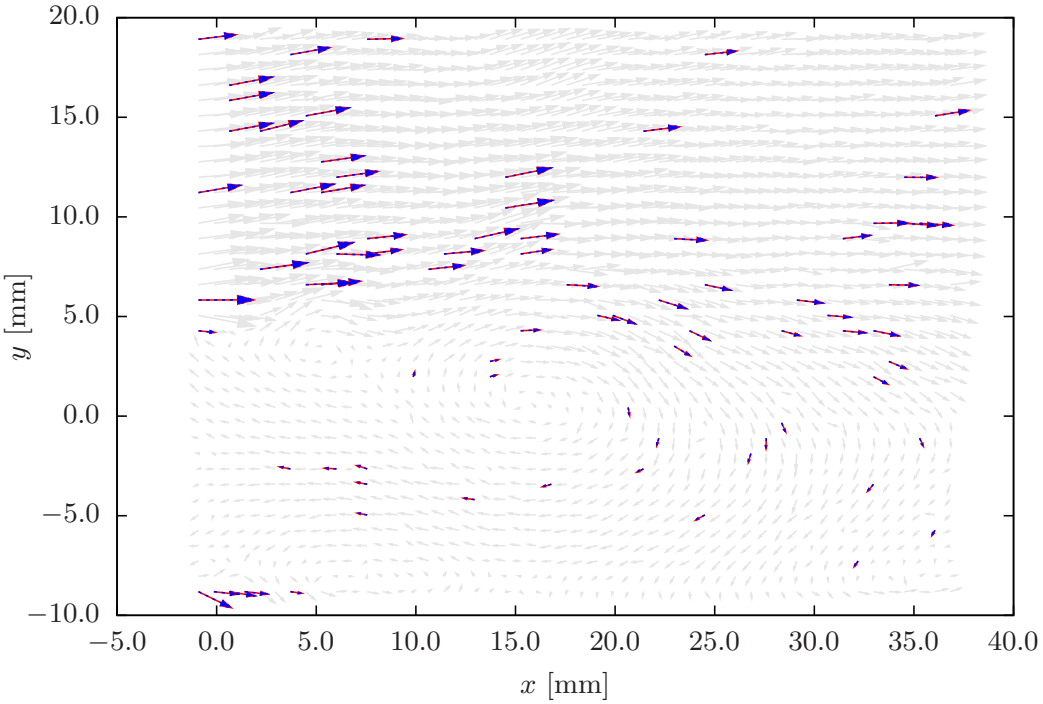


(b) $\boldsymbol{\mu}$ variant methods

Figure 52: 4^{th} flow velocity modes of restored u and v velocity components: $u(q = 40)$, $v(q = 50)$

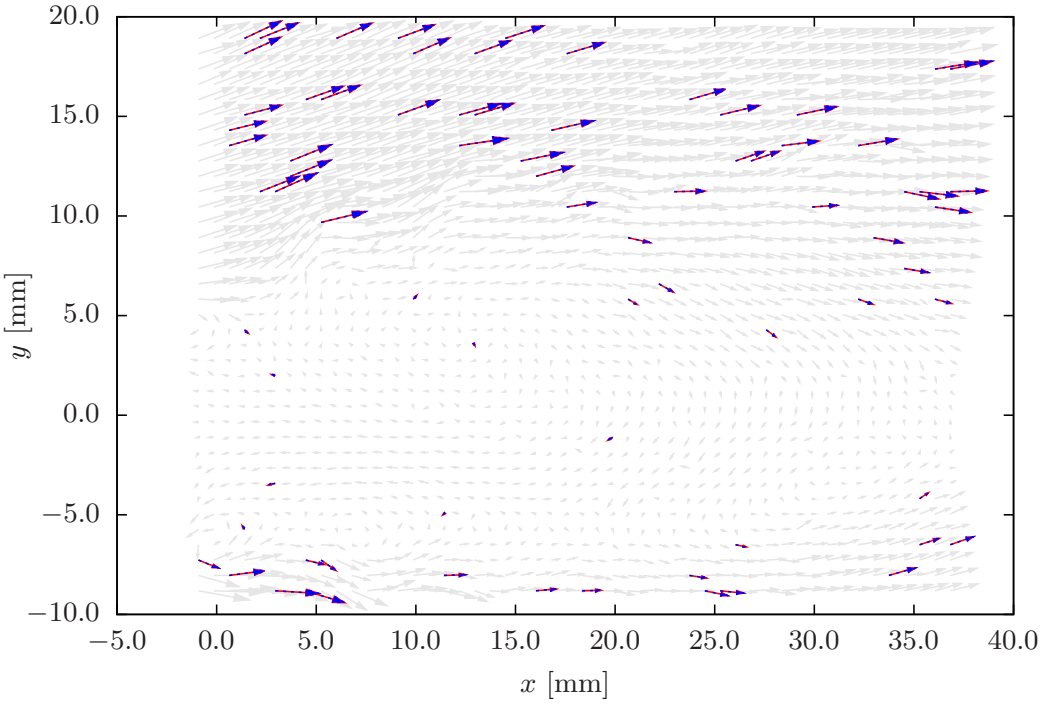


(a) $\boldsymbol{\mu}$ invariant methods

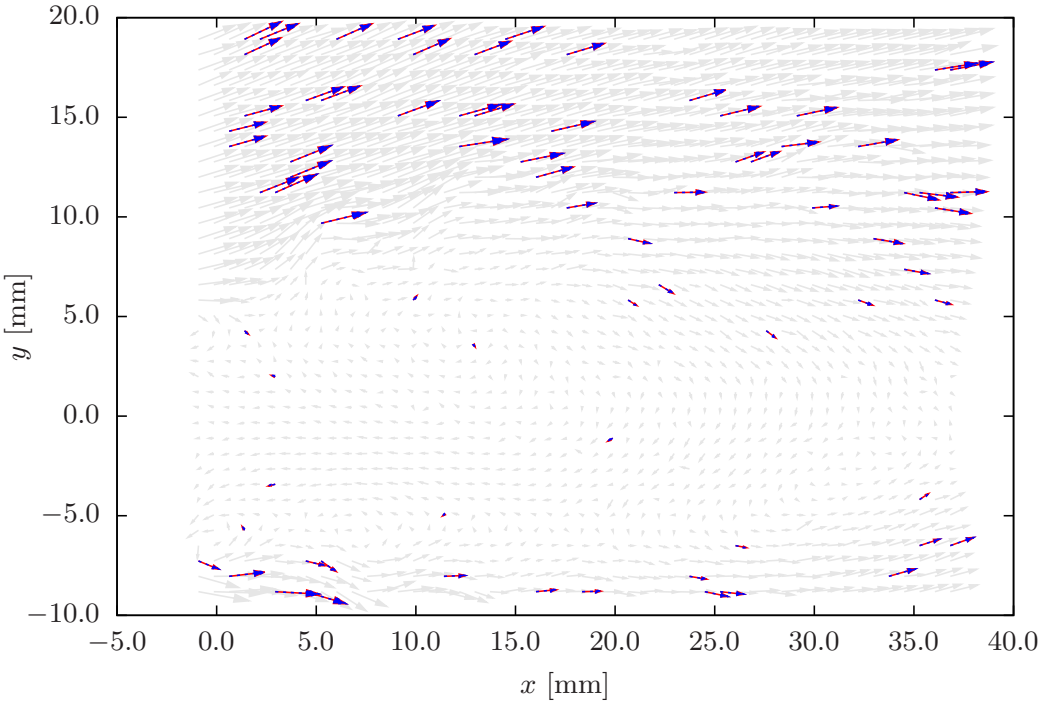


(b) $\boldsymbol{\mu}$ variant methods

Figure 53: Restored 107$^{\text{th}}$ flow velocity snapshot missing 4.32432%: $u(q = 40)$, $v(q = 50)$



(a) $\boldsymbol{\mu}$ invariant methods



(b) $\boldsymbol{\mu}$ variant methods

Figure 54: Restored 100^{th} flow velocity snapshot missing 4.05405%: $u(q = 40)$, $v(q = 50)$

6.3.3 Validation Results

First, Figure 48 delineates the eigenspectrum of the u snapshot ensemble with $q = 40$ in Figures 48(a) and 48(c) and that of the v snapshot ensemble with $q = 50$ in Figures 48(b) and 48(d). As shown in Figure 48, EM-PCA implementations yield eigenspectra clearly identical to those produced by gappy POD implementations for both velocity snapshots. According to the eigenspectra of the u and v snapshot ensembles, $q = 40$ captures nearly 94% of the relative u flow energy, and so does $q = 50$ about 92% of the relative v flow energy. Since the experimented jet flow demonstrates diverse swirling flow patterns after a bluff body, the v snapshot ensemble requires a higher number of modes than the u snapshot ensemble for the same level of relative flow energy.

Analogous to the validation of the eigenspectra in Figure 48, Figures from 49 to 52 compares the first four major modes of the restored velocity fields obtained by gappy POD with those found by EM-PCA implementations. As the observed eigenspectra in Figure 48 show perfect agreements, the flow velocity modes acquired by the EM-PCA well coincide with those by gappy POD regardless of "$\boldsymbol{\mu}$ inv." and "$\boldsymbol{\mu}$ var." implementations. Physically, flow velocity modes represent rudimentary flow structures whose energy strengths are related to their corresponding eigenvalues. For instance, the first flow mode in Figure 49 delineates the most energetic flow pattern, showing the primary flow behavior circulating behind a bluff body. Likewise, other subsequently lower flow velocity modes in Figures from 50 to 52 denote minor flow behavior, capturing low-frequency, locally-dominant vortex flows.

In addition to the validation of the flow velocity modes in Figures from 49 to 52, Figure 53 and Figure 54 show a comparison of the two flow velocity fields restored with both gappy POD and EM-PCA implementations. As an illustration, the 107^{th} and 100^{th} snapshots, both of which lack the most data among snapshots, are portrayed in Figure 53 and Figure 54, respectively. In the two reconstruction cases, the original velocity vectors are in gray, and the restored velocity vectors are in colors according to the implementations. Evidently, both gappy POD and EM-PCA implementations produce restored velocity vectors that are indistinguishable in the case of either their "$\boldsymbol{\mu}$ inv." or "$\boldsymbol{\mu}$ var." implementations. Note that the restored velocity vectors in colors cohesively conform to general flow behavior because

they are estimated based on flow velocity modes.

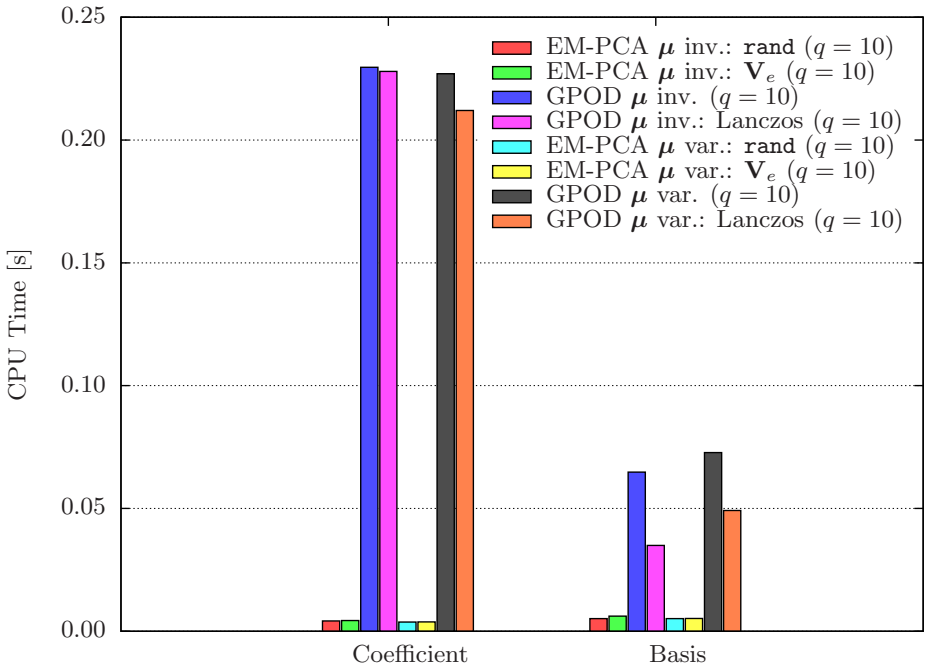
6.4 Numerical Cost Investigation

This section thoroughly examines all the implementations of both gappy POD and the EM-PCA to compare their performance for the application of PIV data restoration. First, it delineates the numerical costs of evaluating each basis and coefficient of both reconstruction methods. Subsequently, it tests all the implementations at low and high q values, measuring their computational times as well as their iteration numbers. Afterwards, it investigates the variations of total evaluation time for all the implementations as q gradually increases. Last, it presents variations in the computational times of the EM-PCA due to random initialization to see if they are comparable with those of gappy POD.

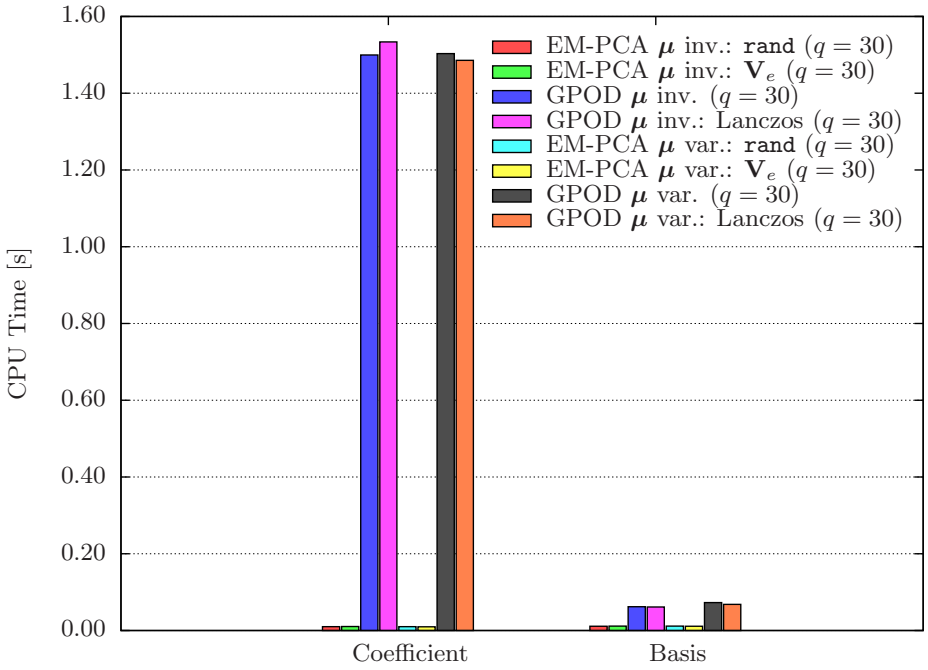
6.4.1 Single Basis and Coefficient Evaluation

The earlier algorithmic analysis in Section 4.1.2.1 suggests that each step of gappy POD requires more effort to evaluate than that of the EM-PCA. In order to substantiate the previous analysis, Lee and Mavris[31] measures the computational time for each basis and coefficient using the u velocity snapshot ensemble at $q = 10$ and $q = 30$. Clearly, both q cases in Figure 55 illustrate that both basis and coefficient evaluation steps of the EM-PCA are considerably more efficient than those of gappy POD. The results of the basis and coefficient evaluations, shown in Figure 55, reveal that the computational time spent on a coefficient evaluation marked by greater than that spent on a basis evaluation.

For instance, Figure 55(a) for $q = 10$ shows that the Lanczos algorithm can save computational time for evaluating the basis of gappy POD; however, it still cannot outperform that of the EM-PCA. Regarding a coefficient evaluation at $q = 10$, gappy POD apparently requires much more computational effort than the EM-PCA. This computational overhead for gappy POD in evaluating a coefficient becomes even greater as q increases from 10 to 30. In Figure 55(b), the $q = 30$ case delineates that the computational time of a coefficient in gappy POD soars dramatically, causing an enormous time differential from that in the EM-PCA. This rapid time increase in the coefficient evaluation of gappy POD indicates that a coefficient evaluation is prone to be a critical computational bottleneck for gappy POD.



(a) $q = 10$



(b) $q = 30$

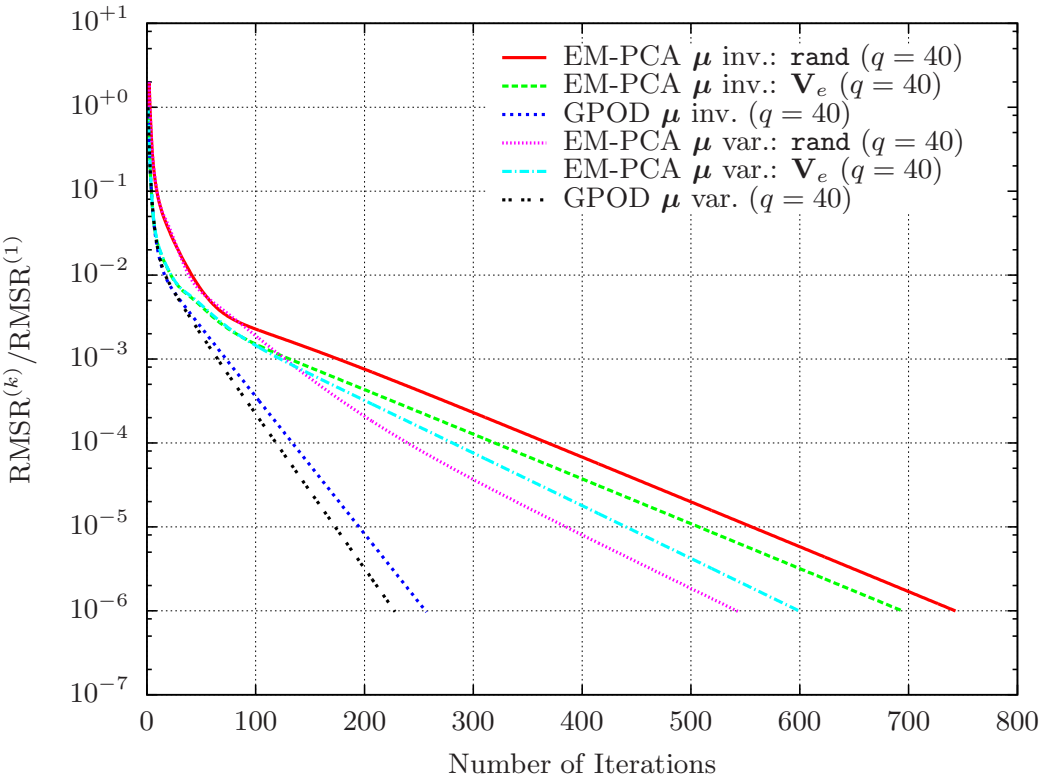
Figure 55: Computational time for a single basis and coefficient evaluation

Unlike the significant disparate coefficient evaluations in Figure 55(b), the basis evaluation of gappy POD at $q = 30$ differs modestly from that of the EM-PCA, resulting in a relatively small computational time difference. Note that the computational benefit of the Lanczos algorithm at $q = 30$ becomes negligible compared to that at $q = 10$ in Figure 55(a). In summary, both basis and coefficient evaluations of the EM-PCA take less time than those of gappy POD, and hence, they are easily scalable with q from a computational aspect. Moreover, for gappy POD, a coefficient evaluation is found to be more computationally demanding than a basis evaluation, especially when q is large.

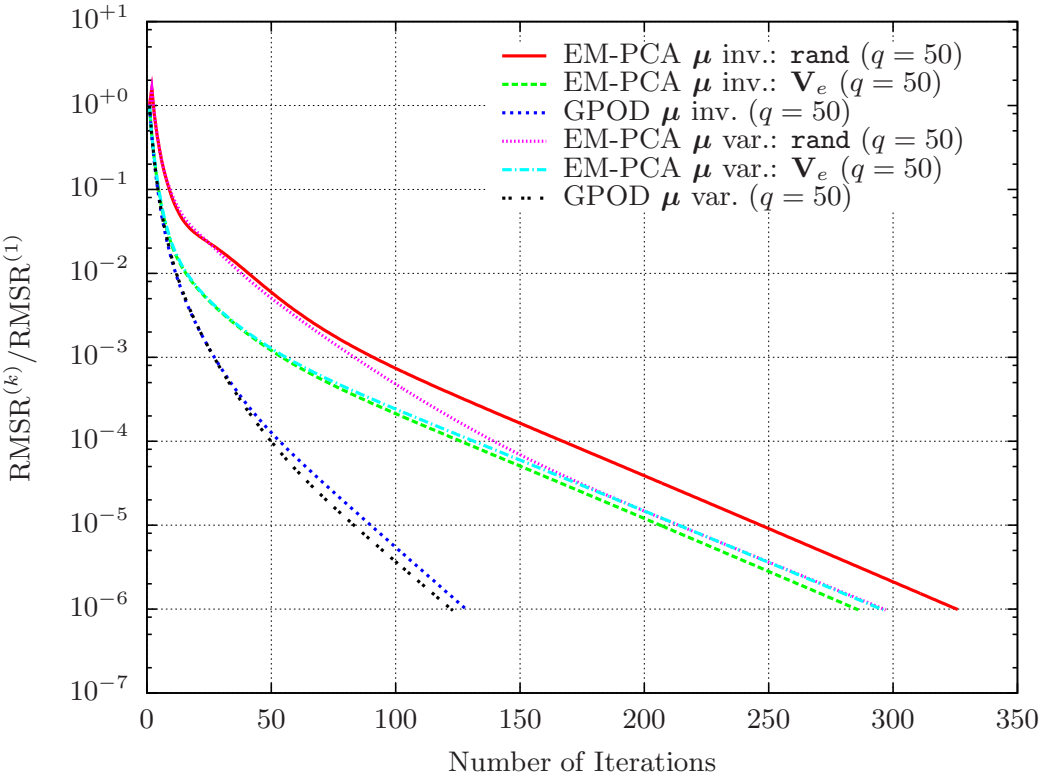
6.4.2 Computational Time Breakdown with the Number of Iterations

As an illustration of the convergence behavior of all the implementations, Figure 56 depicts their convergence histories obtained wiht the u and v snapshot ensembles. In both Figures 56(a) and 56(b), the gappy POD implementations converge generally much faster than the EM-PCA implementations for the same convergence criterion because of the gappy norm. In detail, both gappy POD and EM-PCA implementations exhibit sharp RMSR drops at the the early stage of iterations. However, thereafter, the gappy POD implementations are able to maintain decent convergence rates, whereas the EM-PCA implementations lag due to their relatively sluggish convergence rates. According to Lee and Mavris,[30] the convergence rates of both gappy POD and the EM-PCA are strongly pertinent to their norms; the gappy norm is superior to the L^2 norm in reducing estimation residuals. Note that the EM-PCA implementations with $\mathbf{V}_e$ initialization tend to reach convergence a little earlier than those with random initialization, owing to an informed basis initialization.

In connection with the convergence histories in Figure 56, for further efficiency investigation, Lee and Mavris[31] scrutinized not only total iteration numbers but also the breakdown of total computational times in each evaluation step. To see the performance behavior at both low and high q values, they experimented with all the implementations at two different q values, $q = 5$ and $q = 40$. As an illustration, Figures 57 and 58 delineate the results of $q = 5$ and $q = 40$, respectively. Likewise, the results with the u and v snapshot ensembles are on the top and bottom of Figures 57 and 58. Note that both gappy POD and EM-PCA

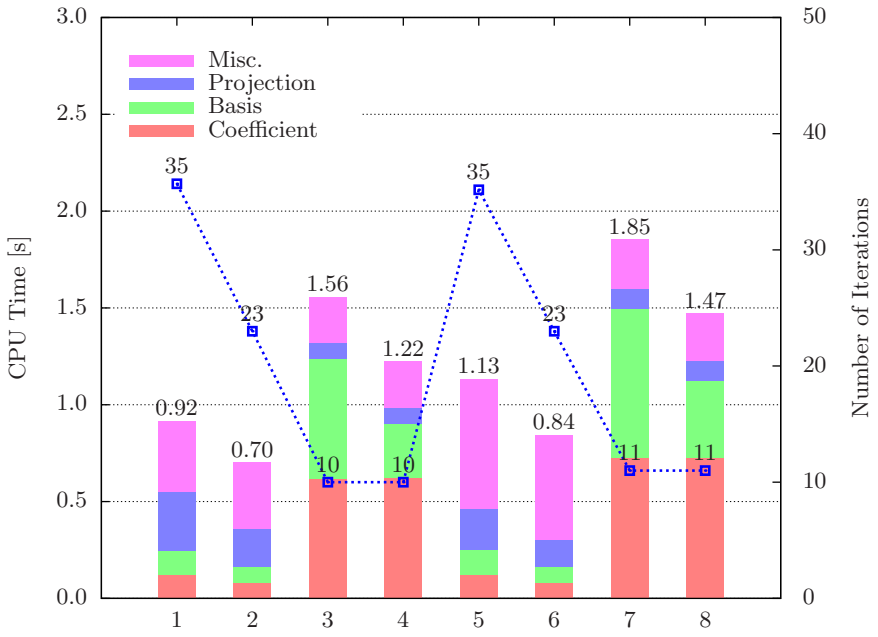


(a) u snapshot ensemble



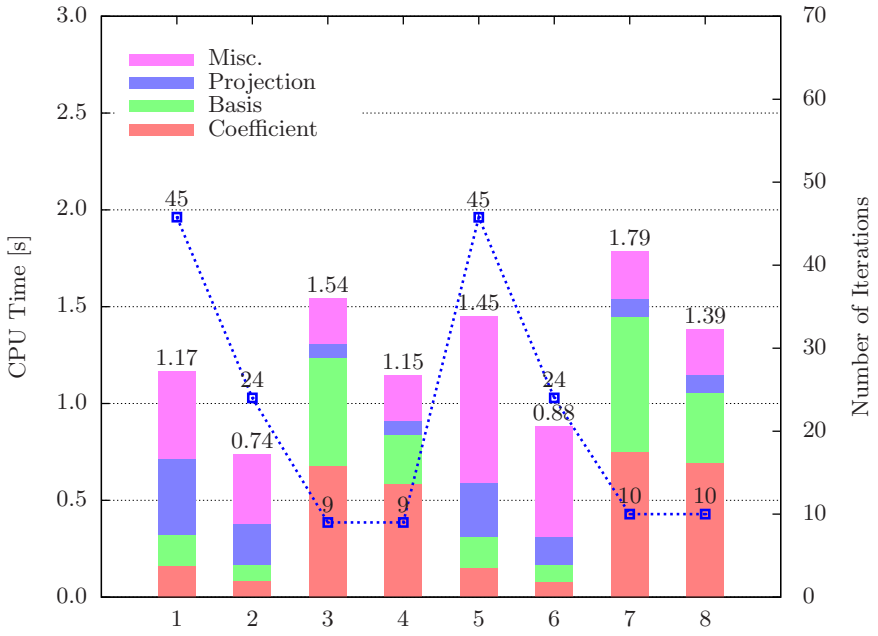
(b) v snapshot ensemble

Figure 56: Convergence histories of the u and v velocity components



1: EM-PCA $\boldsymbol{\mu}$ inv.: `rand` ($q = 5$)
2: EM-PCA $\boldsymbol{\mu}$ inv.: $\mathbf{V}_e$ ($q = 5$)
3: GPOD $\boldsymbol{\mu}$ inv. ($q = 5$)
4: GPOD $\boldsymbol{\mu}$ inv.: Lanczos ($q = 5$)
5: EM-PCA $\boldsymbol{\mu}$ var.: `rand` ($q = 5$)
6: EM-PCA $\boldsymbol{\mu}$ var.: $\mathbf{V}_e$ ($q = 5$)
7: GPOD $\boldsymbol{\mu}$ var. ($q = 5$)
8: GPOD $\boldsymbol{\mu}$ var.: Lanczos ($q = 5$)

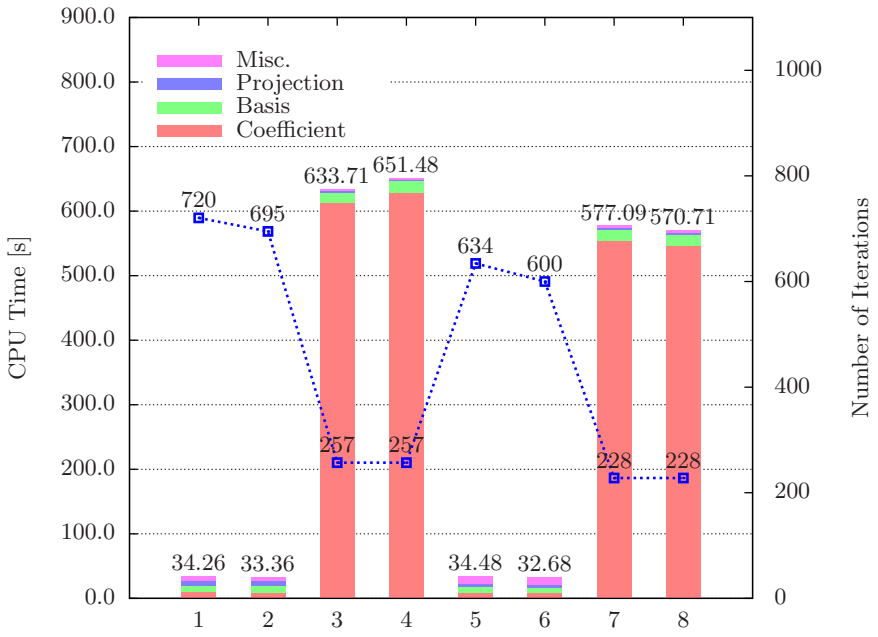
(a) u snapshot ensemble



1: EM-PCA $\boldsymbol{\mu}$ inv.: `rand` ($q = 5$)
2: EM-PCA $\boldsymbol{\mu}$ inv.: $\mathbf{V}_e$ ($q = 5$)
3: GPOD $\boldsymbol{\mu}$ inv. ($q = 5$)
4: GPOD $\boldsymbol{\mu}$ inv.: Lanczos ($q = 5$)
5: EM-PCA $\boldsymbol{\mu}$ var.: `rand` ($q = 5$)
6: EM-PCA $\boldsymbol{\mu}$ var.: $\mathbf{V}_e$ ($q = 5$)
7: GPOD $\boldsymbol{\mu}$ var. ($q = 5$)
8: GPOD $\boldsymbol{\mu}$ var.: Lanczos ($q = 5$)

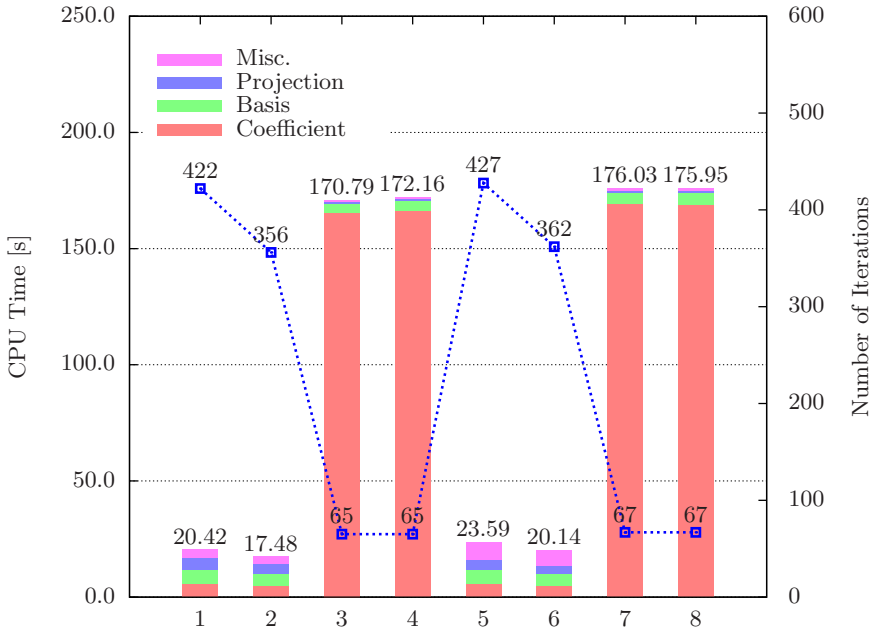
(b) v snapshot ensemble

Figure 57: Computational time decomposition and iteration numbers: $q = 5$



1: EM-PCA $\boldsymbol{\mu}$ inv.: `rand` ($q = 40$)
2: EM-PCA $\boldsymbol{\mu}$ inv.: $\mathbf{V}_e$ ($q = 40$)
3: GPOD $\boldsymbol{\mu}$ inv. ($q = 40$)
4: GPOD $\boldsymbol{\mu}$ inv.: Lanczos ($q = 40$)
5: EM-PCA $\boldsymbol{\mu}$ var.: `rand` ($q = 40$)
6: EM-PCA $\boldsymbol{\mu}$ var.: $\mathbf{V}_e$ ($q = 40$)
7: GPOD $\boldsymbol{\mu}$ var. ($q = 40$)
8: GPOD $\boldsymbol{\mu}$ var.: Lanczos ($q = 40$)

(a) u snapshot ensemble



1: EM-PCA $\boldsymbol{\mu}$ inv.: `rand` ($q = 40$)
2: EM-PCA $\boldsymbol{\mu}$ inv.: $\mathbf{V}_e$ ($q = 40$)
3: GPOD $\boldsymbol{\mu}$ inv. ($q = 40$)
4: GPOD $\boldsymbol{\mu}$ inv.: Lanczos ($q = 40$)
5: EM-PCA $\boldsymbol{\mu}$ var.: `rand` ($q = 40$)
6: EM-PCA $\boldsymbol{\mu}$ var.: $\mathbf{V}_e$ ($q = 40$)
7: GPOD $\boldsymbol{\mu}$ var. ($q = 40$)
8: GPOD $\boldsymbol{\mu}$ var.: Lanczos ($q = 40$)

(b) v snapshot ensemble

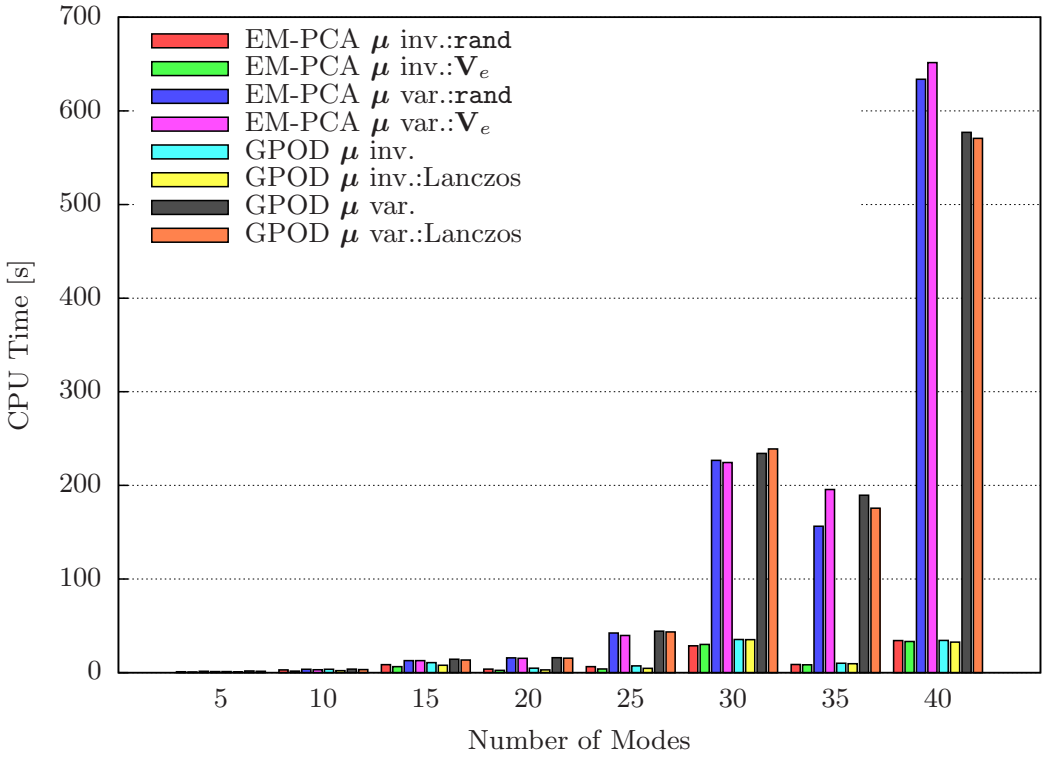
Figure 58: Computational time decomposition and iteration numbers: $q = 40$

implementations result in total computational times that are inversely correlated with their total iteration numbers; namely, gappy POD implementations are slower despite their lower iteration numbers, and vice versa for EM-PCA implementations.

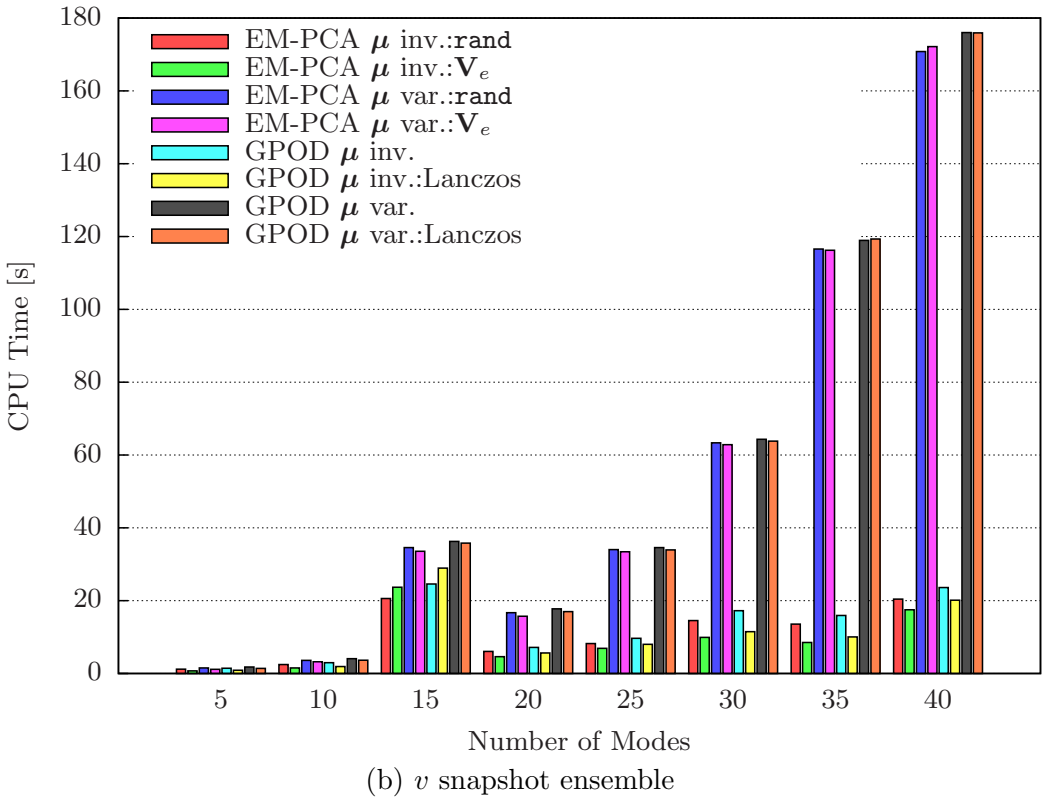
For example, in the case of $q = 5$, both Figures 57(a) and 57(b) show that the overall computational time differentials between gappy POD and EM-PCA implementations are relatively small. Note that half of the total time of gappy POD implementations is spent on evaluating coefficients, implying that a coefficient evaluation may decelerate gappy POD as q increases. The Lanczos algorithm is beneficial to gappy POD implementations as it reduces the basis evaluation time, but its effect on saving a total time is insignificant. In contrast to the $q = 5$ case, Figure 58 for $q = 40$ shows that the advantage of the EM-PCA to gappy POD, previously small in Figure 57, is now noticeably substantial. As anticipated from the $q = 30$ case in Section 6.4.1, the gappy POD implementations are mainly overwhelmed by their coefficient evaluations. Since a coefficient evaluation is a main obstacle to the performance of gappy POD, the Lanczos algorithm is no more useful at enhancing the performance of gappy POD. Unlike gappy POD implementations, EM-PCA implementations take a relatively equal amount of time for basis and coefficient evaluations, even at a large q value.

6.4.3 Performance Variations with the Increase of the Number of Modes

In order to accomplish exhaustive computational performance investigation, Lee and Mavris[31] tested all the implementations, changing q from 5 to 40 at intervals of 5. As shown in Figure 59, computational time measured with the u and v snapshot ensembles are delineated in Figure 59(a) and Figure 59(b), respectively. In Figure 59, both gappy POD and EM-PCA implementations demonstrate completely disparate variations in their computational times with respect to q increments; gappy POD implementations are easily affected by the q increase, but EM-PCA implementations are not. More specifically, the gappy POD implementations start to suffer significantly from $q = 30$ in Figure 59(a) for the u snapshot ensemble, as it does from $q = 25$ in Figure 59(b) for the v snapshot ensemble. By



(a) u snapshot ensemble



(b) v snapshot ensemble

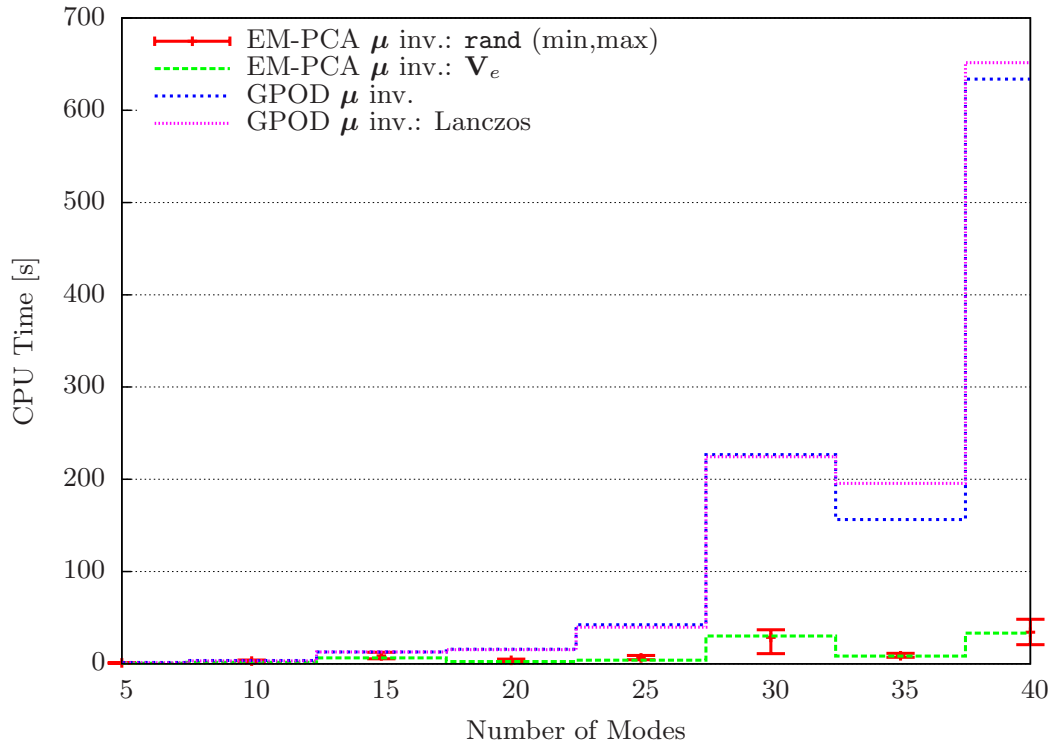
Figure 59: Computational time variations with q changes

contrast, the EM-PCA implementations generally exhibit gradual computational time increases with q regardless of the velocity snapshot ensembles. Note that the behavior of the computational times is not completely linear with q for either the gappy POD or EM-PCA implementations. For example, all the implementations show surges in their computational times at certain q values such as $q = 30$ and $q = 15$ for the u and v snapshot ensembles, respectively. Overall, the gappy POD implementations are clearly not as scalable as the EM-PCA implementations with the q rise.

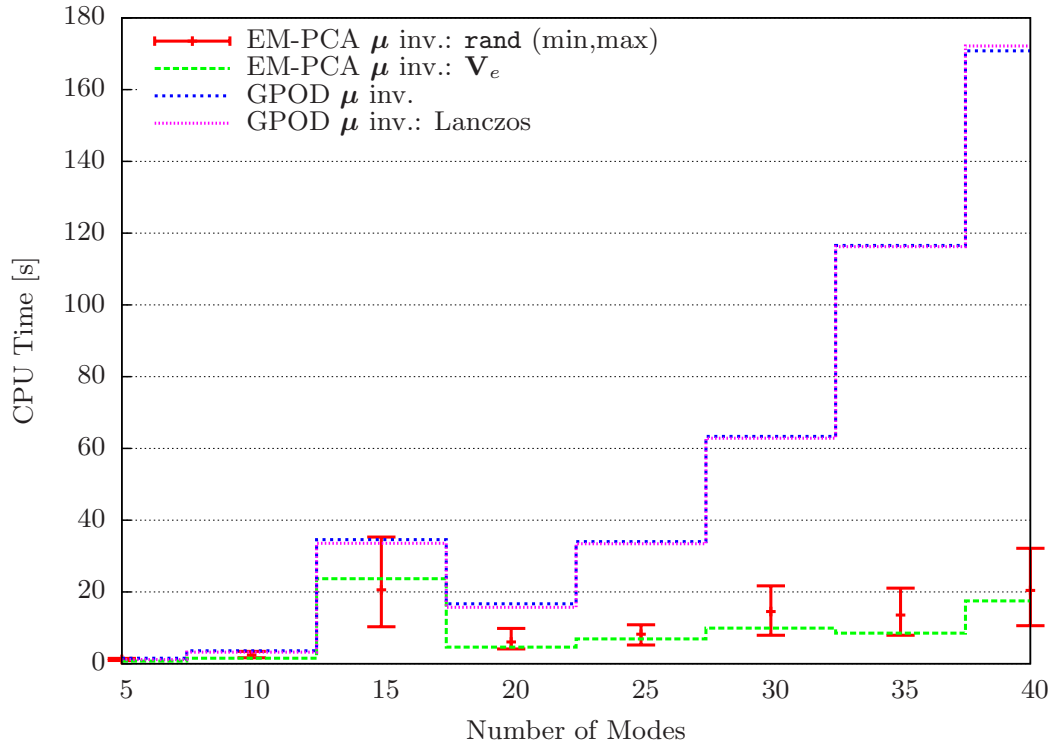
6.4.4 Effect of Random Initialization for the EM-PCA

To corroborate the preceding investigation of the computational time variations in Figure 59, Lee and Mavris[31] examined variations in the total times of the EM-PCA implementations due to random initialization. Figure 60 delineates the upper and lower bounds of the computational times of the EM-PCA implementations measured for 100 random initializations at the q interval of 5, similar to Figure 59. In addition, Figure 60 depicts the computational times of other gappy POD implementations along with the EM-PCA implementation with the $\mathbf{V}_e$ initialization for performance comparison purposes. In Figure 60, the computational times of the $\boldsymbol{\mu}$ invariant implementations are in Figures 60(a) and 60(b), and those of the $\boldsymbol{\mu}$ variant implementations are in Figures 60(c) and 60(d). Figure 60 also arranges the comparison results with the u and v snapshot ensembles on the top and bottom, respectively.

As shown in Figure 60, the randomly initialized EM-PCA implementation overall outperforms the other implementations. In the case of the EM-PCA implementation with random initialization, its computational time variations are small with the u snapshot ensemble in Figures 60(a) and 60(c), but they are noticeable with the v snapshot ensemble in Figures 60(b) and 60(d) at certain q values such as $q = 15$, $q = 30$, $q = 35$, and $q = 40$. Moreover, because of random initialization, the EM-PCA implementations occasionally take more time than gappy POD implementations, for example, in the case of $q = 15$ for the v snapshot ensemble in Figures 60(b) and 60(d). Although the EM-PCA implementation exhibits irregular computational performance with random initialization, it is consistently

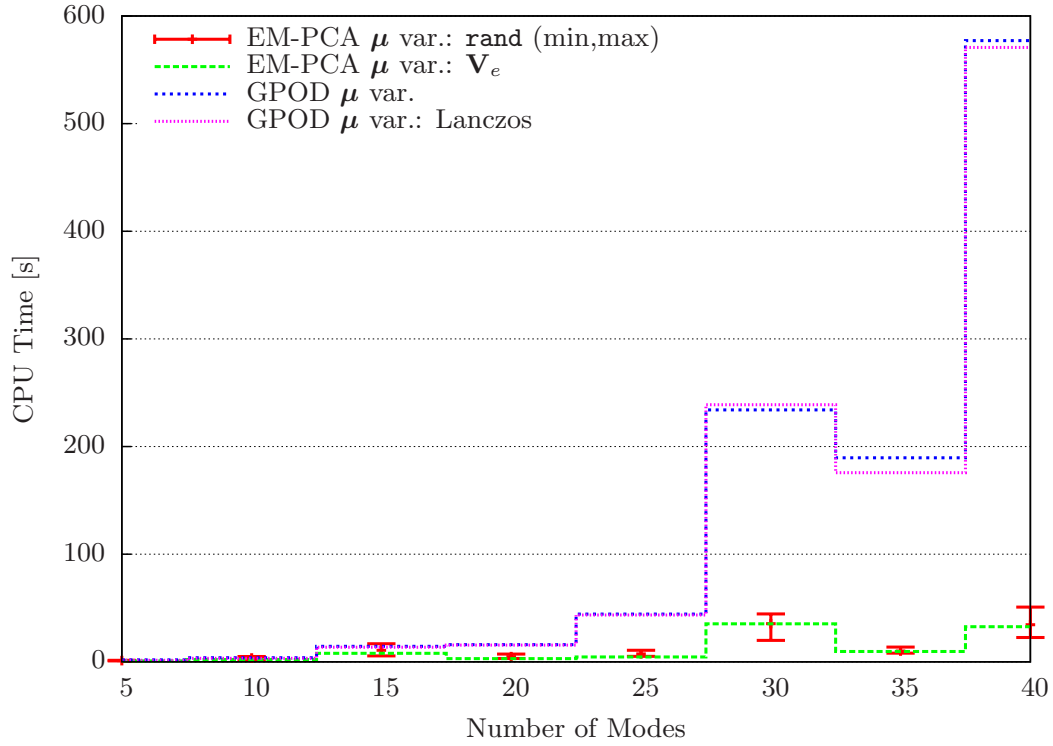


(a) u snapshot ensemble with the $\boldsymbol{\mu}$ invariant methods

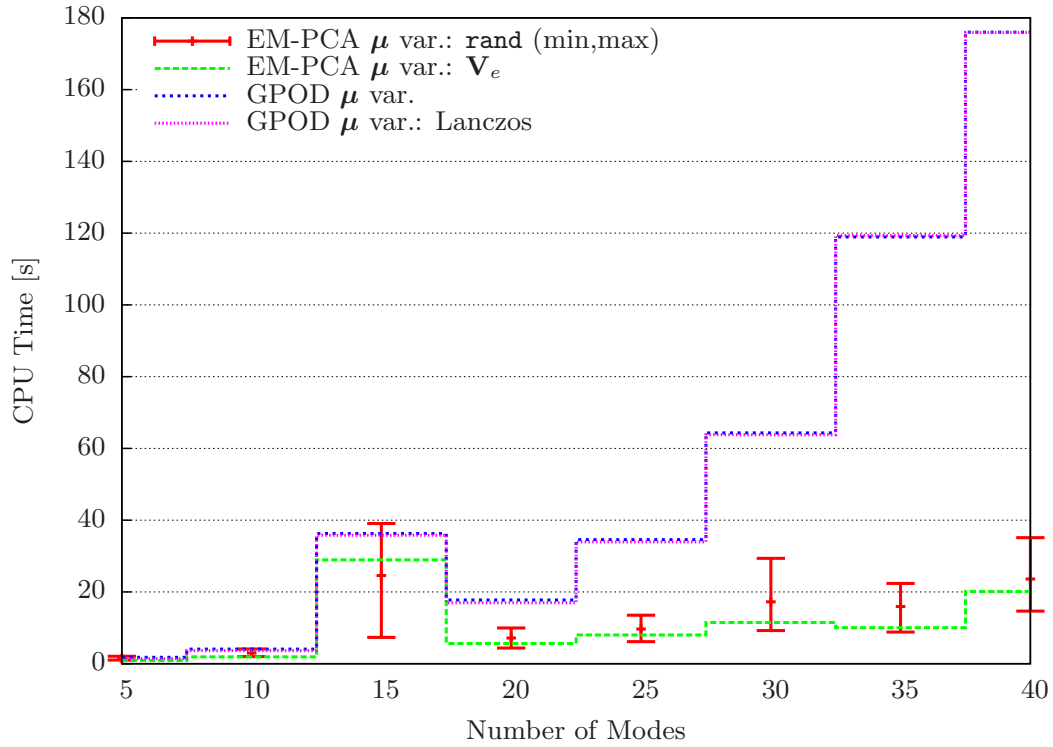


(b) v snapshot ensemble with the $\boldsymbol{\mu}$ invariant methods

Figure 60: Computational time variations of "EM-PCA `rand` init." with q changes



(c) u snapshot ensemble with the $\boldsymbol{\mu}$ variant methods



(d) v snapshot ensemble with the $\boldsymbol{\mu}$ variant methods

Figure 60: Computational time variations of "EM-PCA `rand` init." with q changes

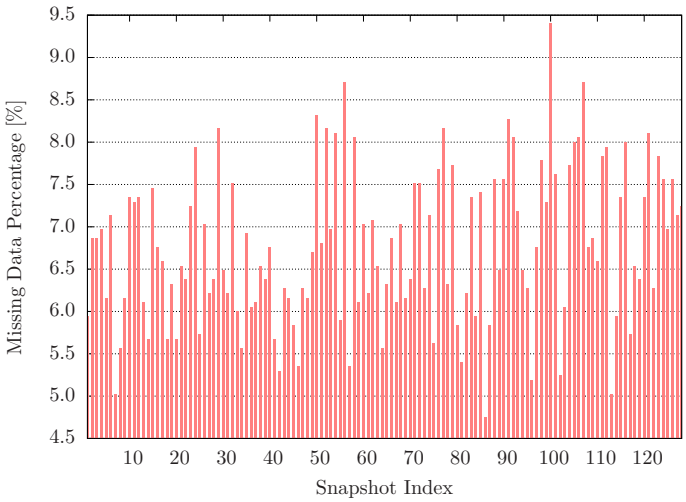
superior to the gappy POD implementation throughout q changes, regardless of the snapshot ensembles. Therefore, Figure 60 shows that the EM-PCA is more computationally efficient than gappy POD in general, and the $\mathbf{V}_e$ initialization is preferable to the EM-PCA in a conservative sense. Note that the Lanczos algorithm barely benefits the gappy POD implementations for computational time saving, and sometimes it performs even worse in such large q value cases as $q = 35$ and $q = 40$ for the u snapshot ensemble in Figure 60(a).

6.5 PIV Data Restoration with Artificially Missing Data

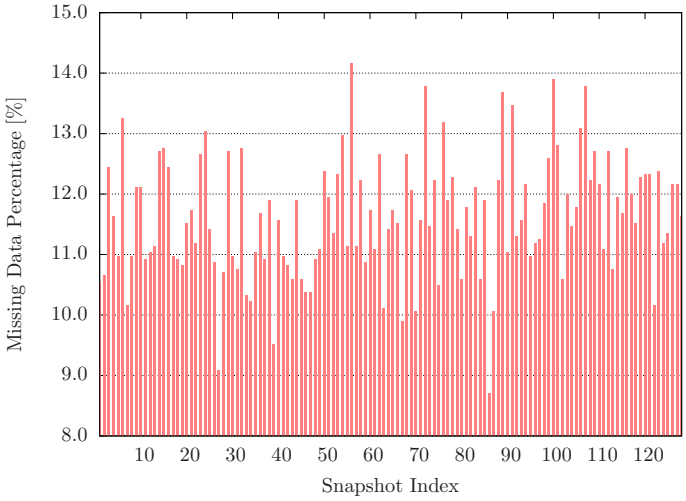
To further examine the reconstruction capability of the EM-PCA compared to that of gappy POD, this research devised three PIV data sets by artificially eliminating some data from the PIV data set used in Chapter 6. With the original PIV data set, whose rate of unreliable data is 1.7867%, this research attempted to randomly remove 5%, 10%, and 15% of the PIV measurements, resulting in additional PIV data sets whose missing percentages are 6.7508%, 11.6043%, and 16.5431%, respectively. As an illustration, Figure 61 delineates the distributions of missing data rates across snapshots for the three PIV data sets.

6.5.1 Convergence Histories

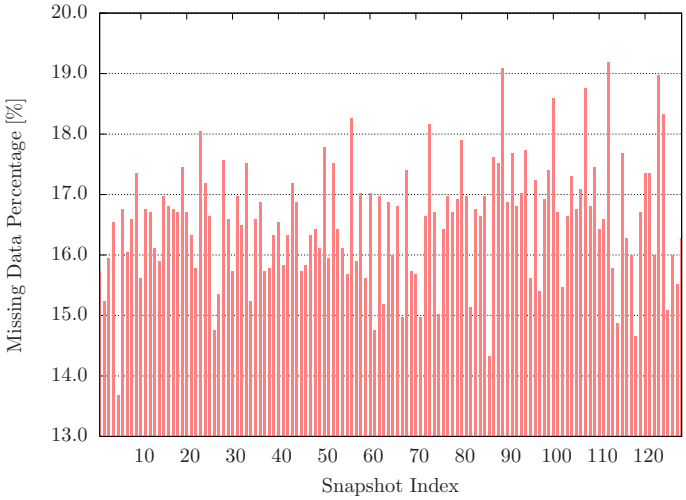
Given the three randomly marred PIV data sets, this research utilized the implementations of gappy POD and the EM-PCA as it did for the original PIV data set in Section 6.3.1. In Figures 62 to 64, the convergence behavior of both algorithms is considerably similar to that observed for the original PIV data set, depicted in Figure 56. Overall, the EM-PCA takes more iterations than gappy POD regardless of their "$\boldsymbol{\mu}$ inv./$\boldsymbol{\mu}$ var." implementations and the three PIV data sets. Note that Figure 64(a) shows that the "$\boldsymbol{\mu}$ inv." implementations of both gappy POD and the EM-PCA suffer from significantly poor convergence performance, and they are terminated by exceeding the preset maximum number of iterations $50,000$ in restoring the u snapshot ensemble, which has 16.5431% of its data is absent. As the rate of missing data gradually increases from 6.7508% to 16.5431%, the implementations require higher numbers of iterations, implying wider computational performance gaps between gappy POD and the EM-PCA.



(a) Overall missing data percentage: 6.7508%

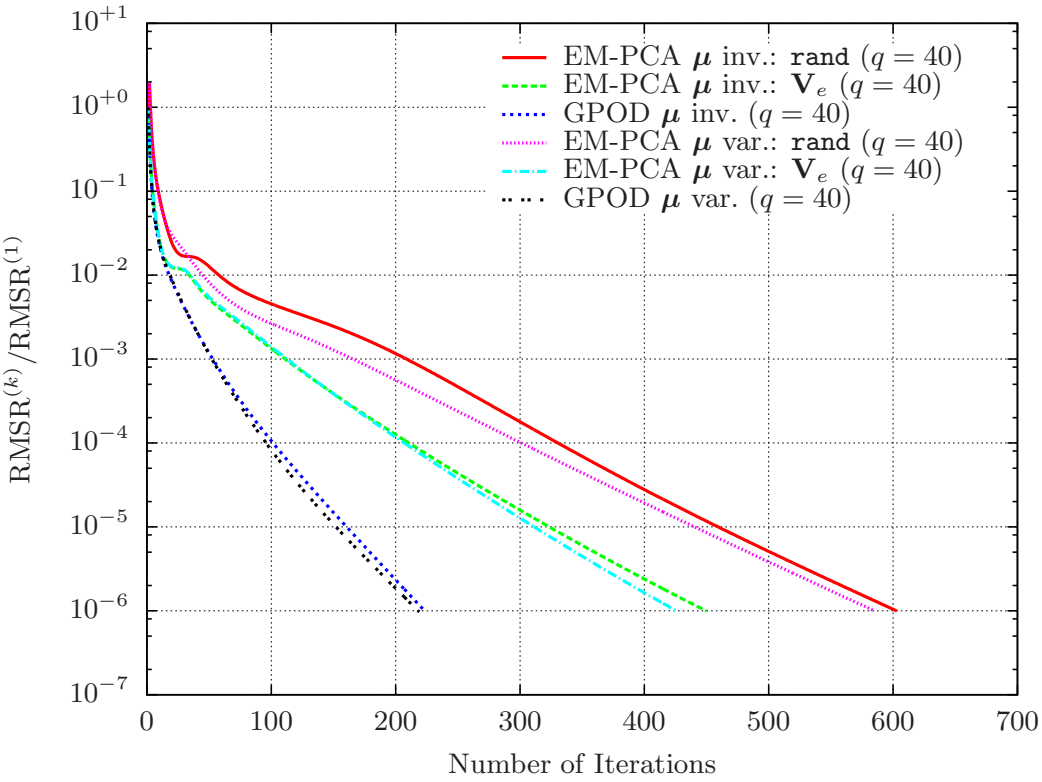


(b) Overall missing data percentage: 11.6043%

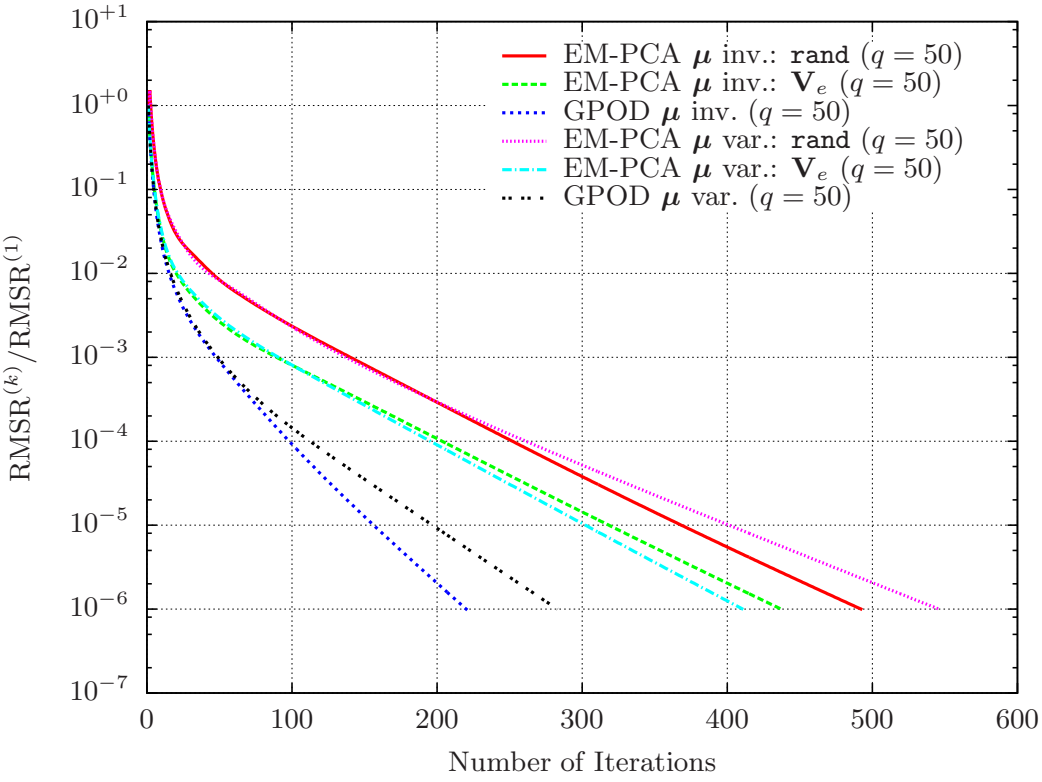


(c) Overall missing data percentage: 16.5431%

Figure 61: Missing data rates of randomly marred PIV data sets

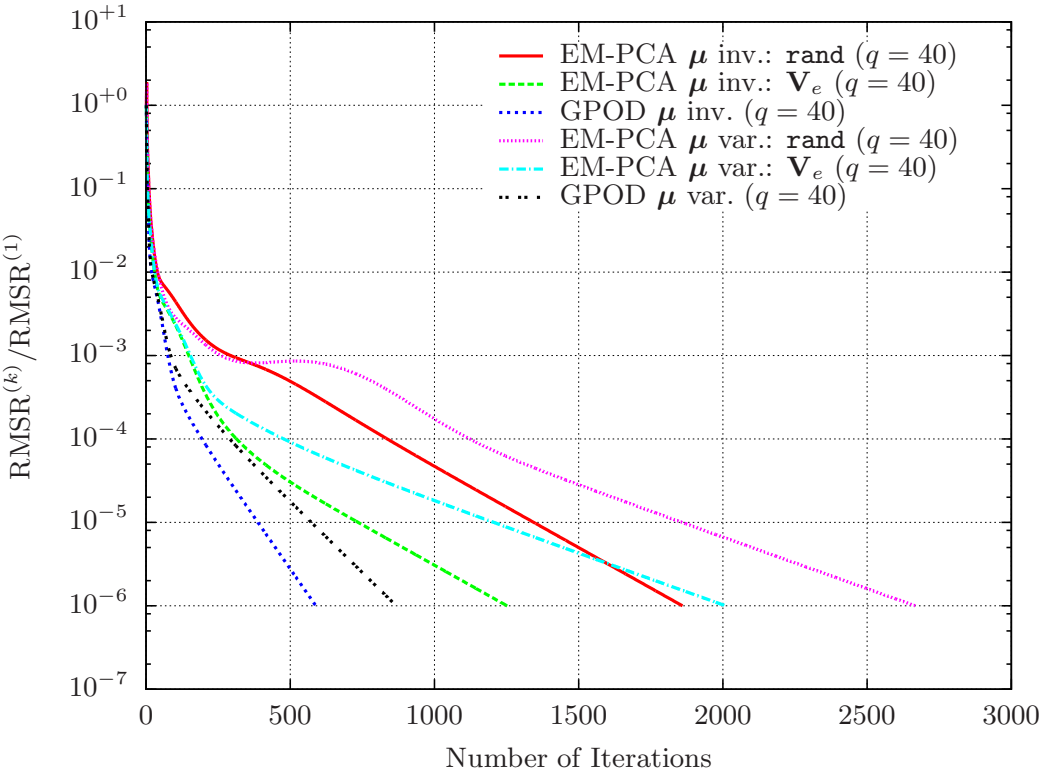


(a) u snapshot ensemble

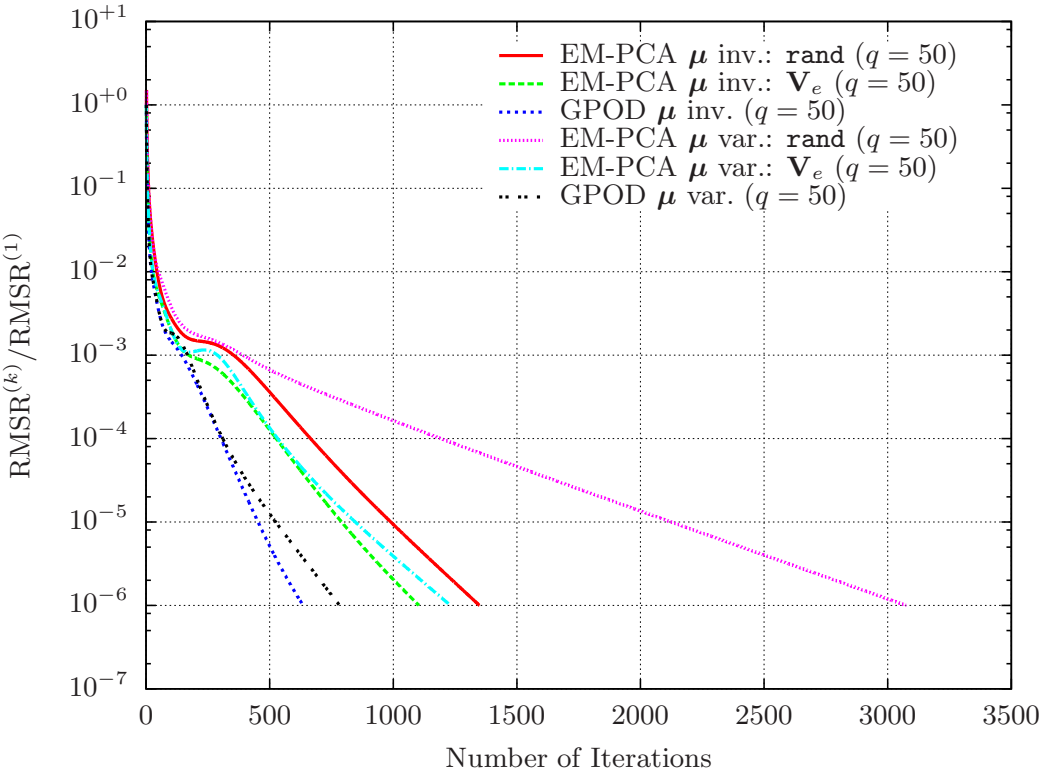


(b) v snapshot ensemble

Figure 62: Convergence histories of the u and v velocity components (6.7508% missing)

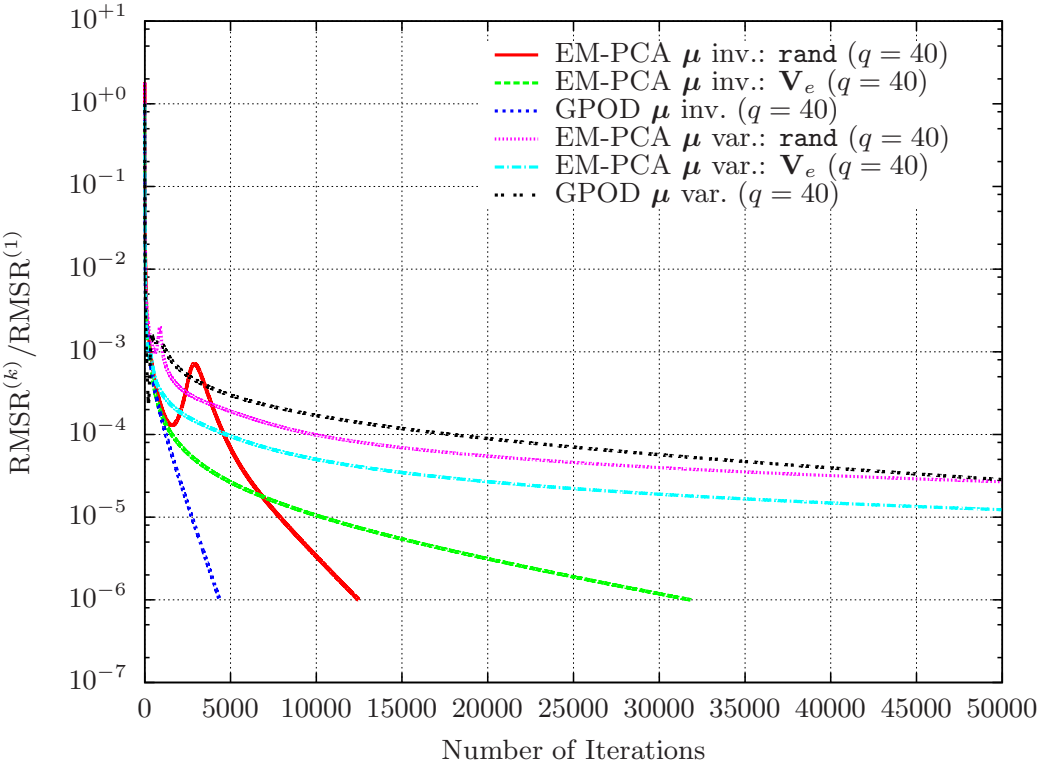


(a) u snapshot ensemble

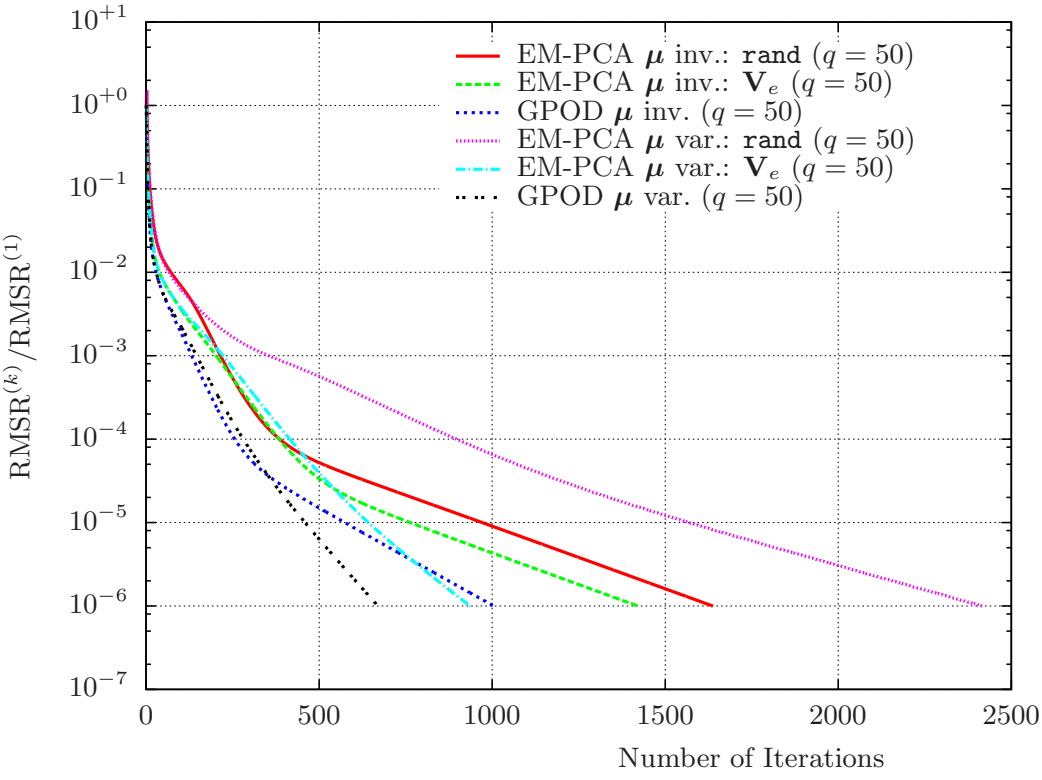


(b) v snapshot ensemble

Figure 63: Convergence histories of the u and v velocity components (11.6043% missing)



(a) u snapshot ensemble



(b) v snapshot ensemble

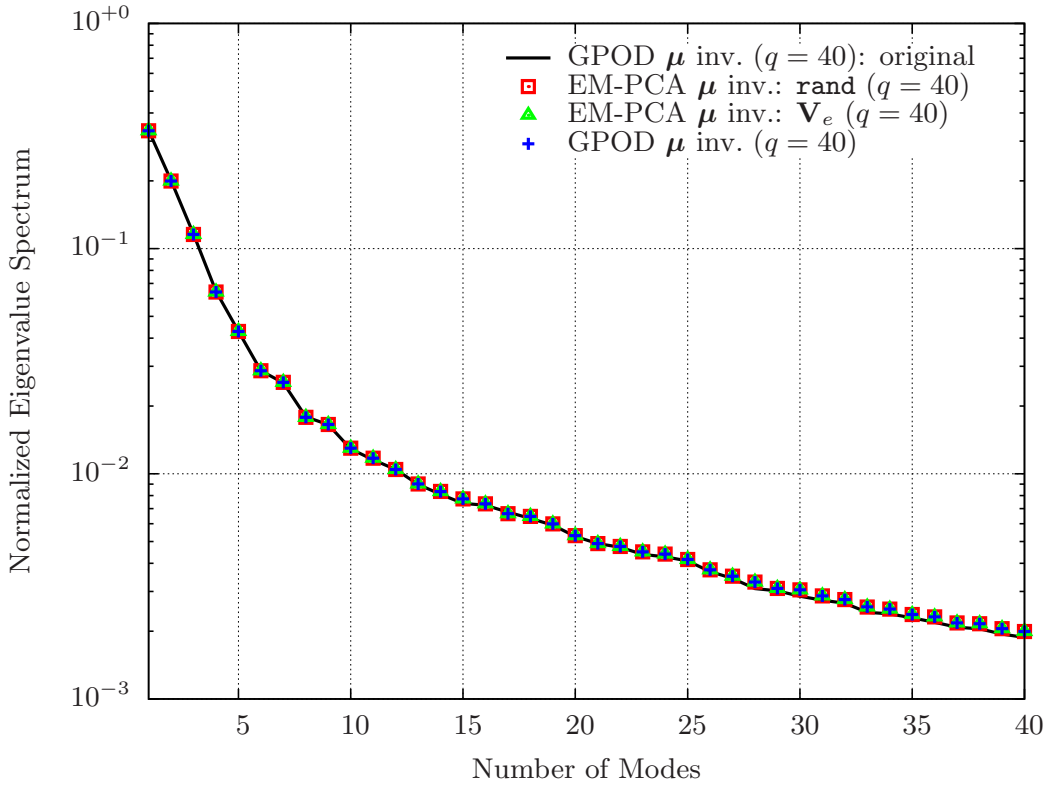
Figure 64: Convergence histories of the u and v velocity components (16.5431% missing)

6.5.2 Validation Results

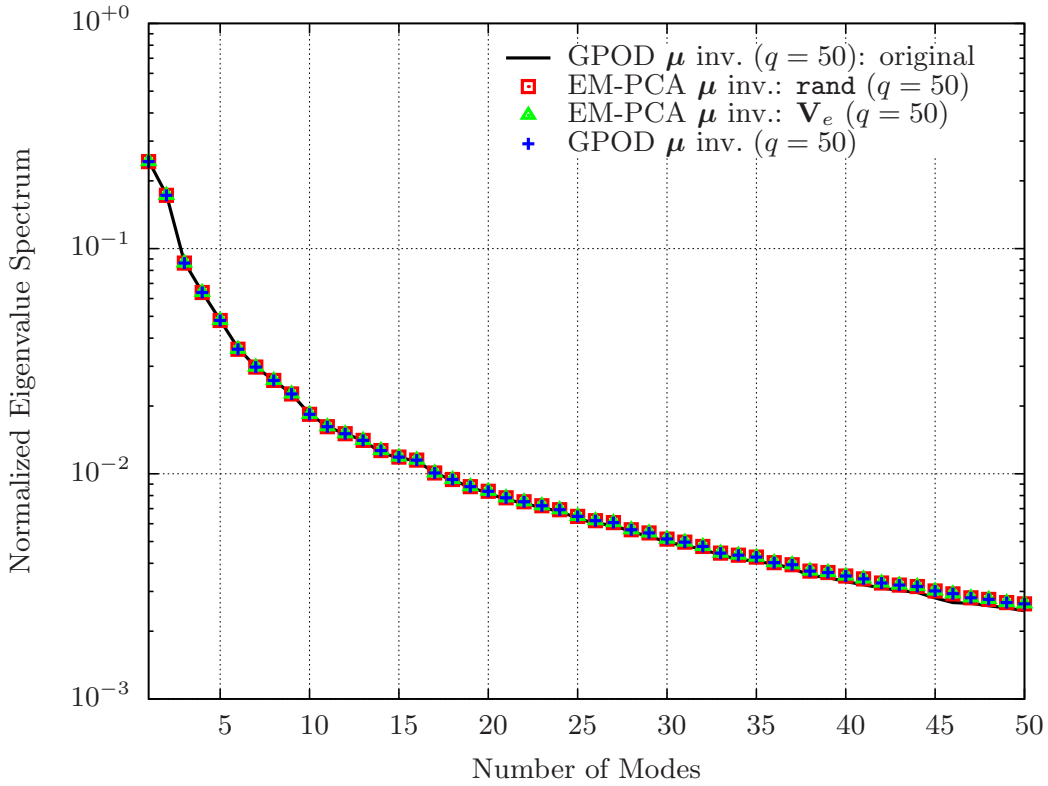
To test both the EM-PCA and gappy POD with the three synthetically generated PIV data sets, this research evaluates their restoration qualities in terms of a normalized eigenspectrum and a restored velocity field. As shown in Figures 65 to 67, the EM-PCA yields normalized eigenvalues that are identical to those obtained by gappy POD except for the u snapshot ensemble, which lacks 16.5431% of its data. In particular, Figure 67(a) shows that "EM-PCA $\boldsymbol{\mu}$ inv.: `rand`" more accurately locates dominant eigenvalues than other implementations, and Figure 67(c) shows that the EM-PCA implementations produce eigenvalues closer to those of the original u snapshot ensemble than gappy POD even though all "$\boldsymbol{\mu}$ inv." implementations equally struggle for convergence, as delineated in Figure 64(a). Note that normalized eigenvalues exhibit more discrepancies than those of the original snapshot ensembles at higher modes as a PIV data set contains more missing data.

Similar to the validation results of normalized eigenspectrum, restored velocity fields at the 107^{th} and 100^{th} snapshots are illustrated in Figures 68 to 73. In general, the EM-PCA repairs absent velocity vectors the same as gappy POD, and the reproduced velocity vectors by both the EM-PCA and gappy POD algorithms accurately match the original velocity vectors. However, both reconstruction algorithms result in misaligned velocity vectors with the original velocity vectors in the right regions behind a bluff body where the jet flow is highly swirling. For instance, Figure 71 shows that both the EM-PCA and gappy POD overestimate a velocity vector at the coordinate $(0.0, 5.0)$, and likewise, Figure 72 shows that both algorithms yield a misleading velocity vector at the very same coordinate. Note that the restored velocity vector at $(0.0, 5.0)$ in Figure 72(b) is considerably inaccurate because of poor convergence histories noticed in Figure 64(a).

In summary, this research observed that the EM-PCA is able to provide at least the same reconstruction results as gappy POD, and in certain cases, such as the u snapshot ensemble missing 16.5431% of its data, it produces more accurate results than gappy POD as delineated in Figure 67(a).

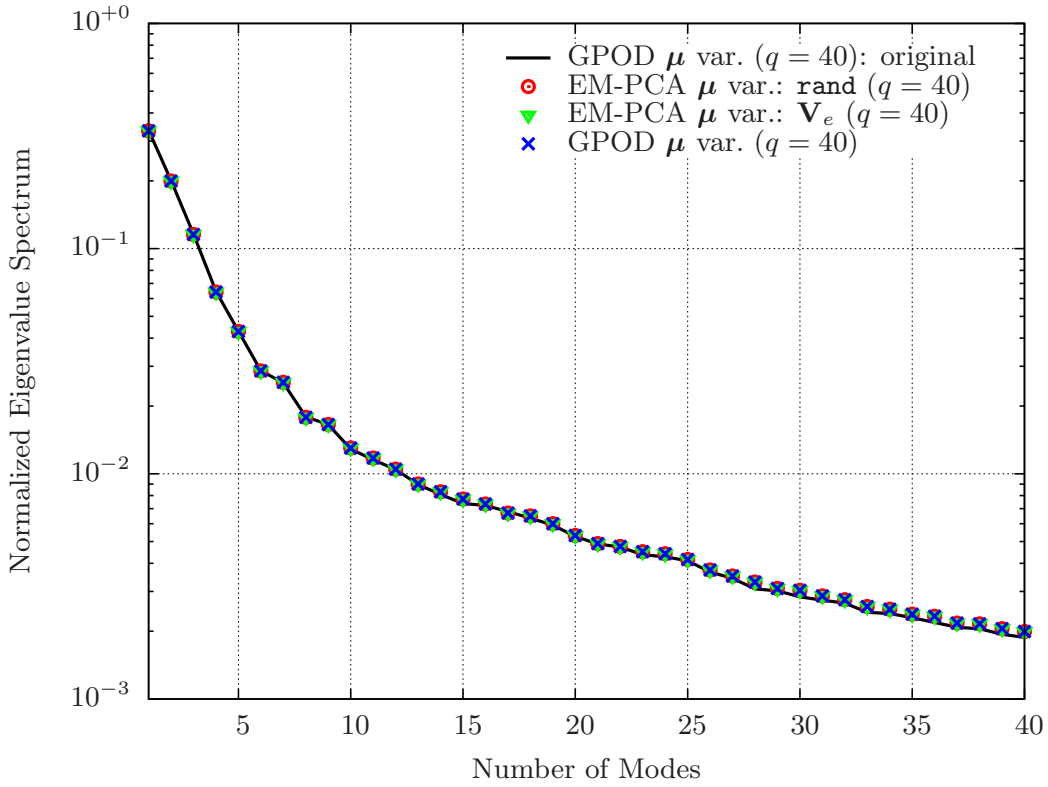


(a) u snapshot ensemble

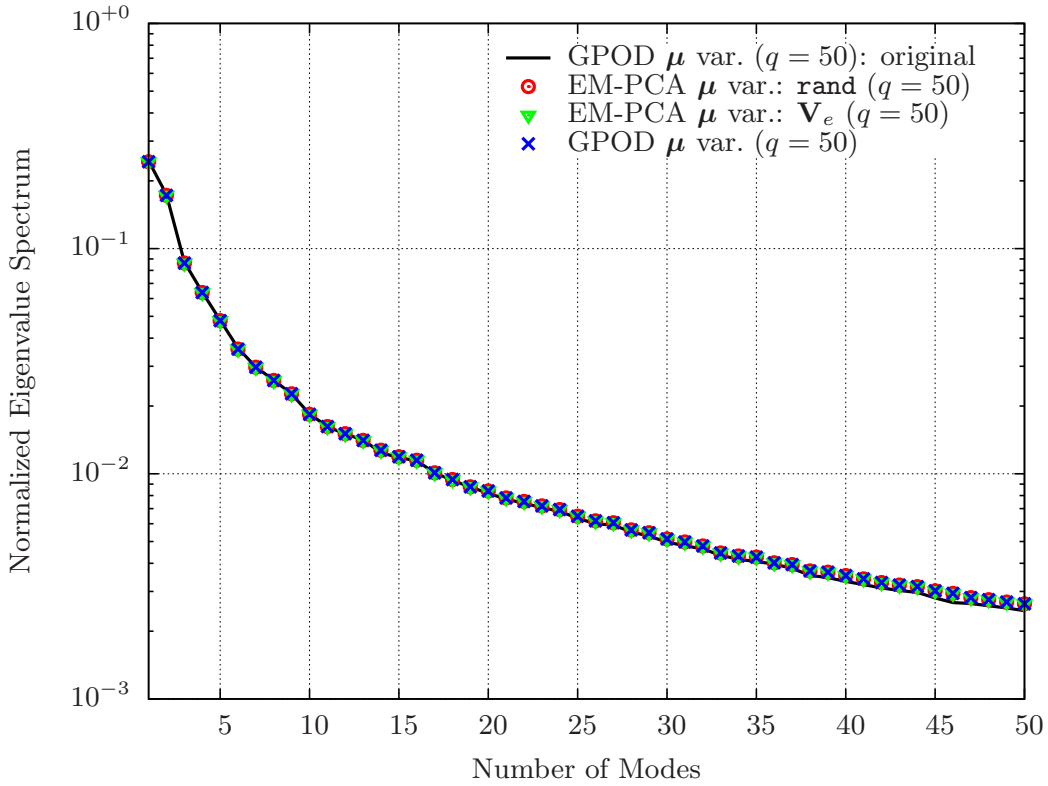


(b) v snapshot ensemble

Figure 65: Eigenspectra of restored u and v velocity components (6.7508% missing)

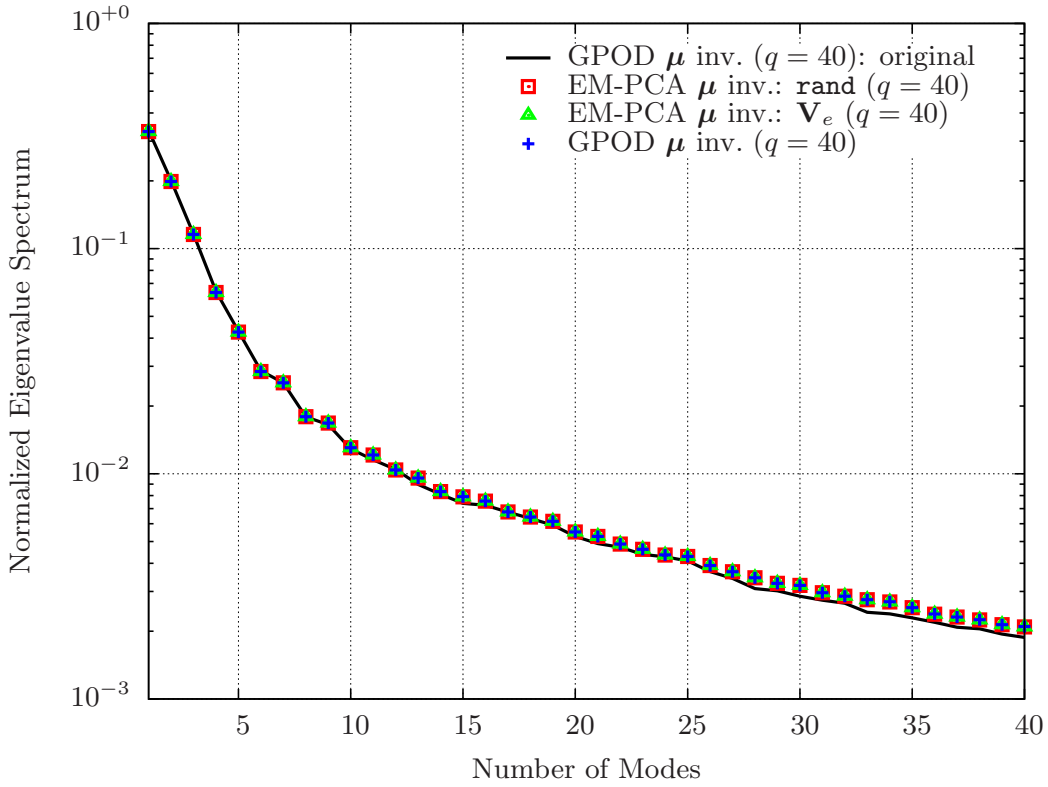


(c) u snapshot ensemble

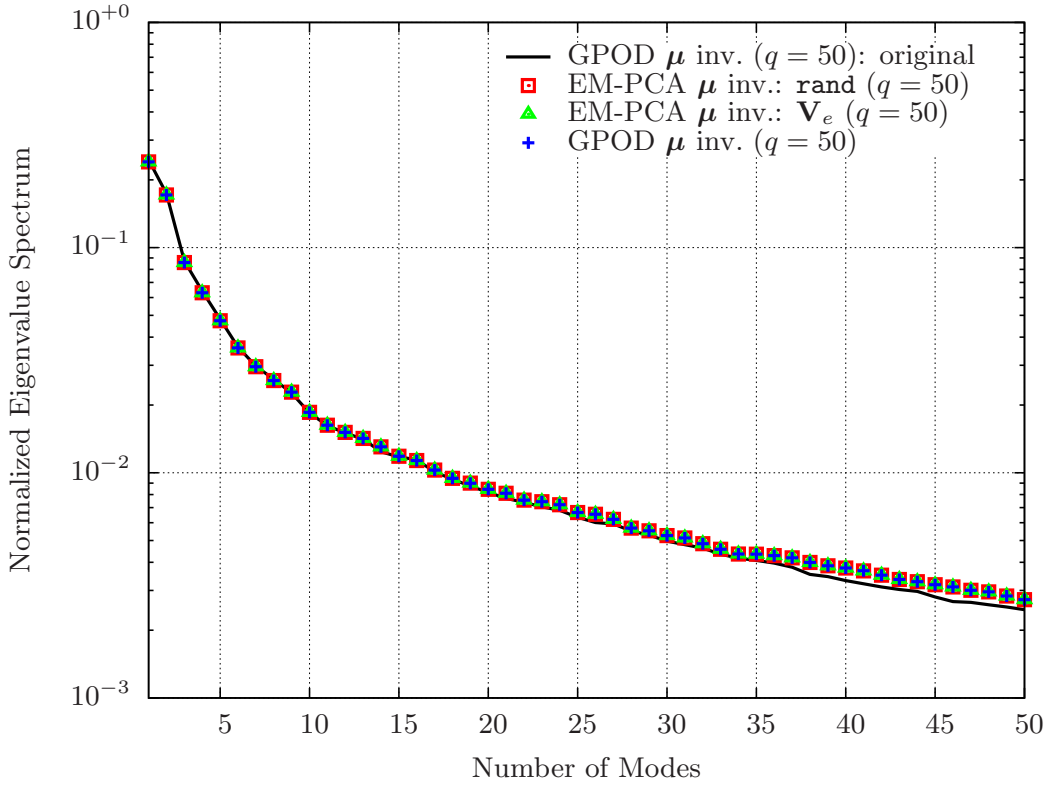


(d) v snapshot ensemble

Figure 65: Eigenspectra of restored u and v velocity components (6.7508% missing)

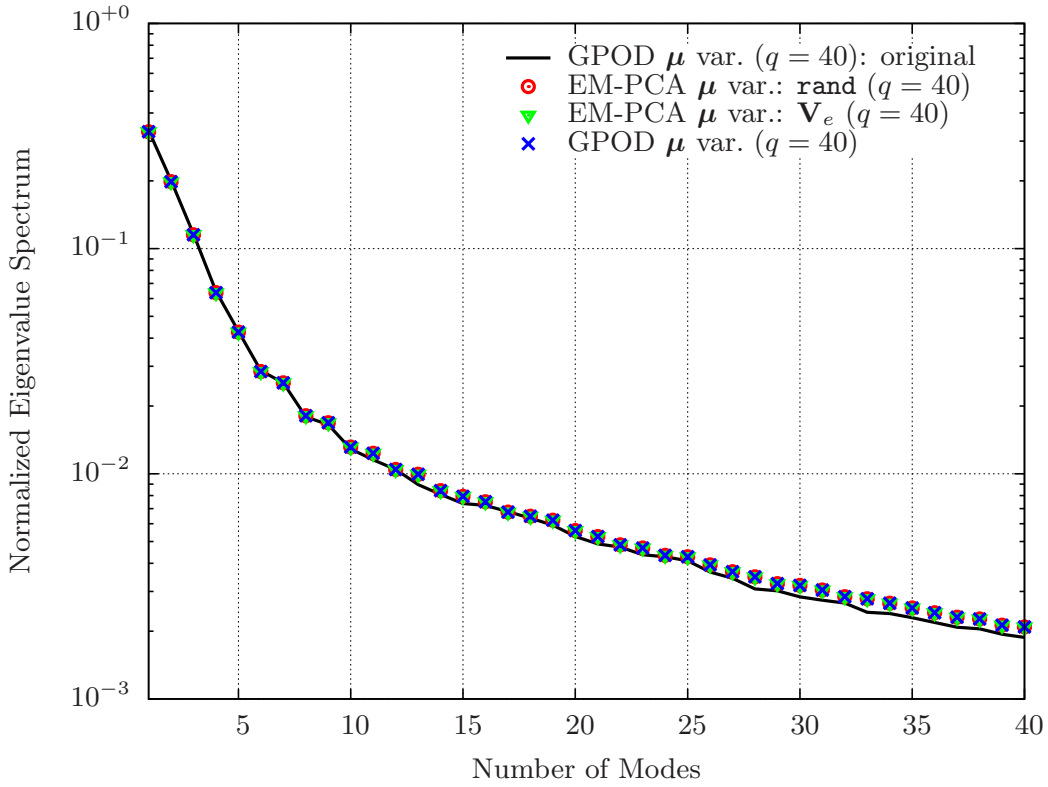


(a) u snapshot ensemble

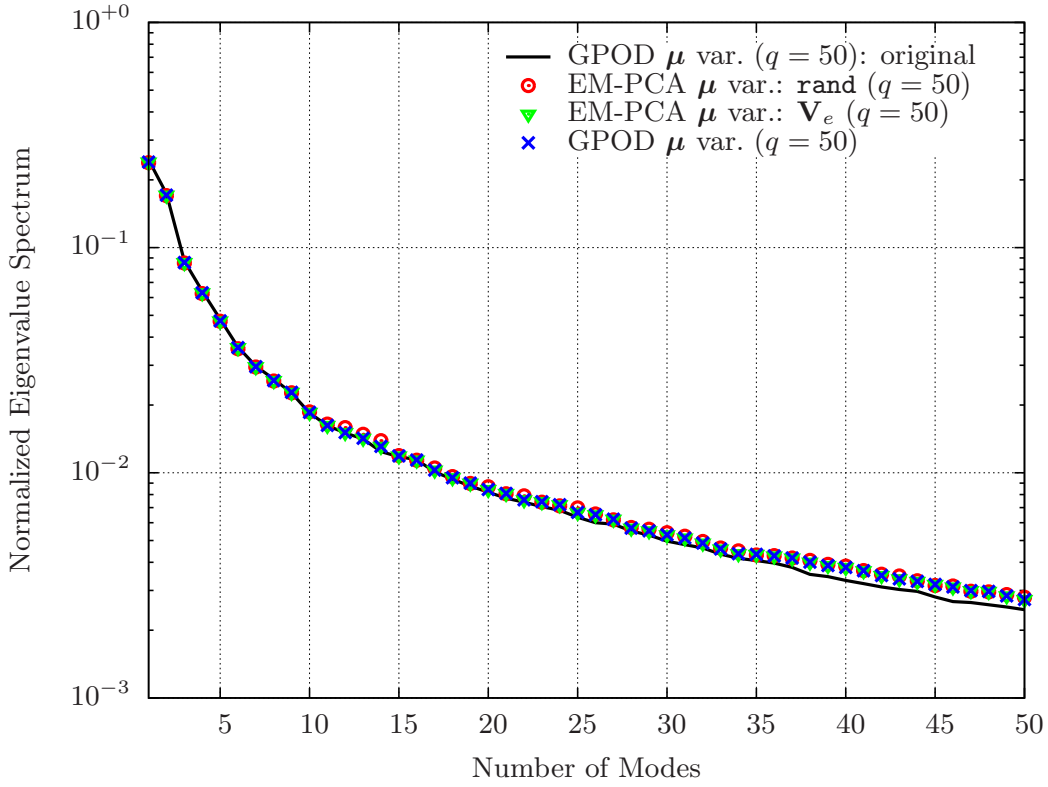


(b) v snapshot ensemble

Figure 66: Eigenspectra of restored u and v velocity components (11.6043% missing)

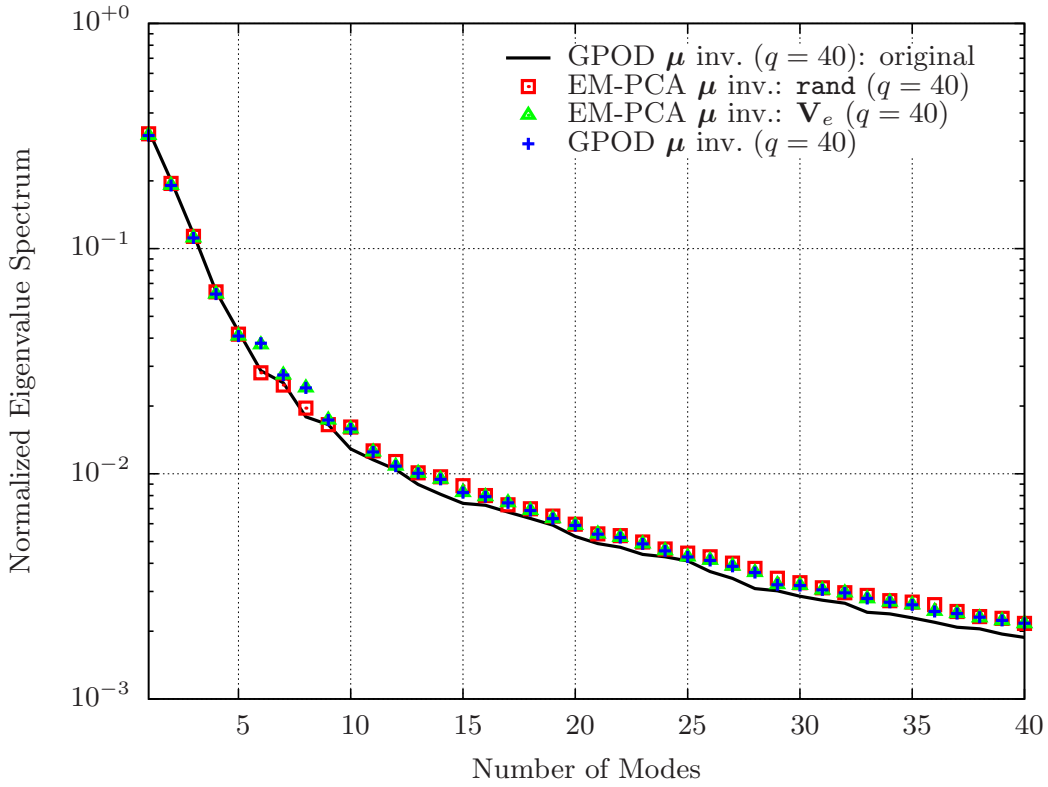


(c) u snapshot ensemble

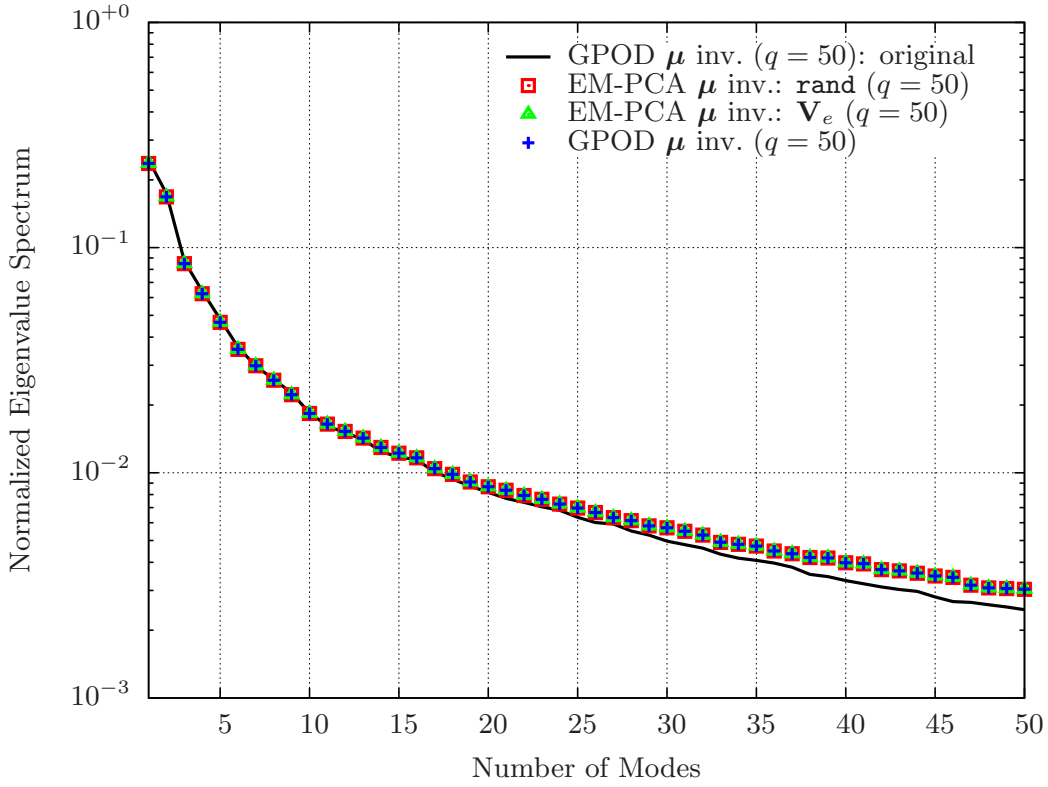


(d) v snapshot ensemble

Figure 66: Eigenspectra of restored u and v velocity components (11.6043% missing)

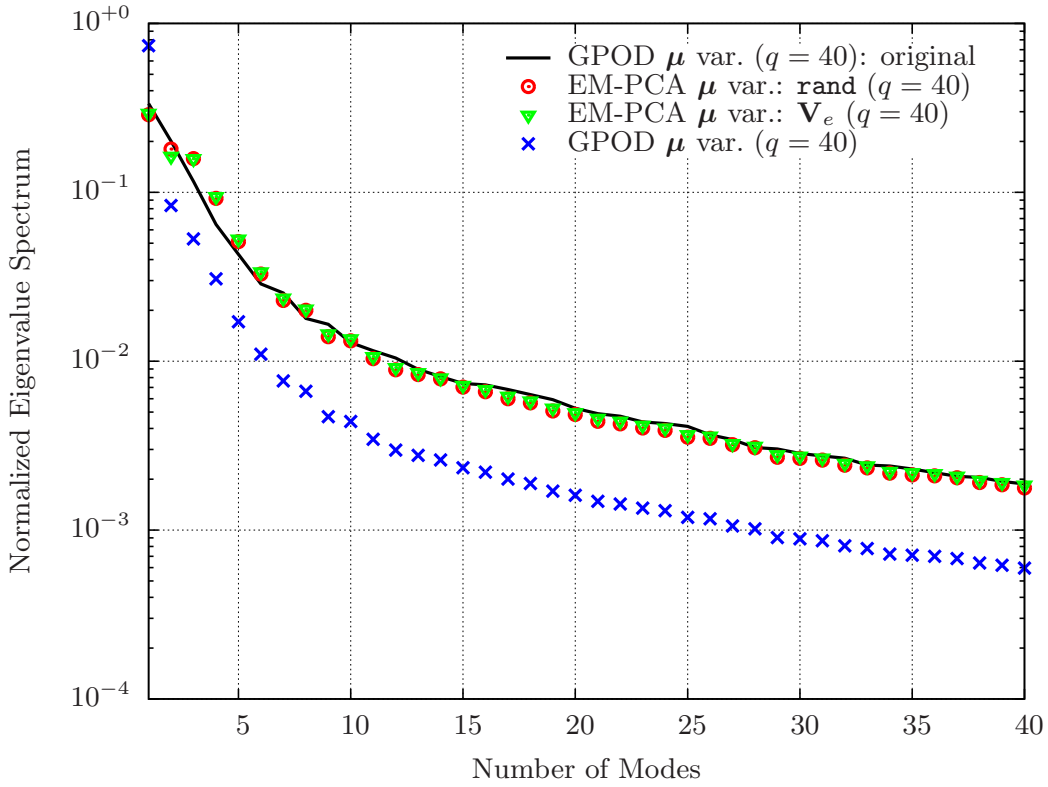


(a) u snapshot ensemble

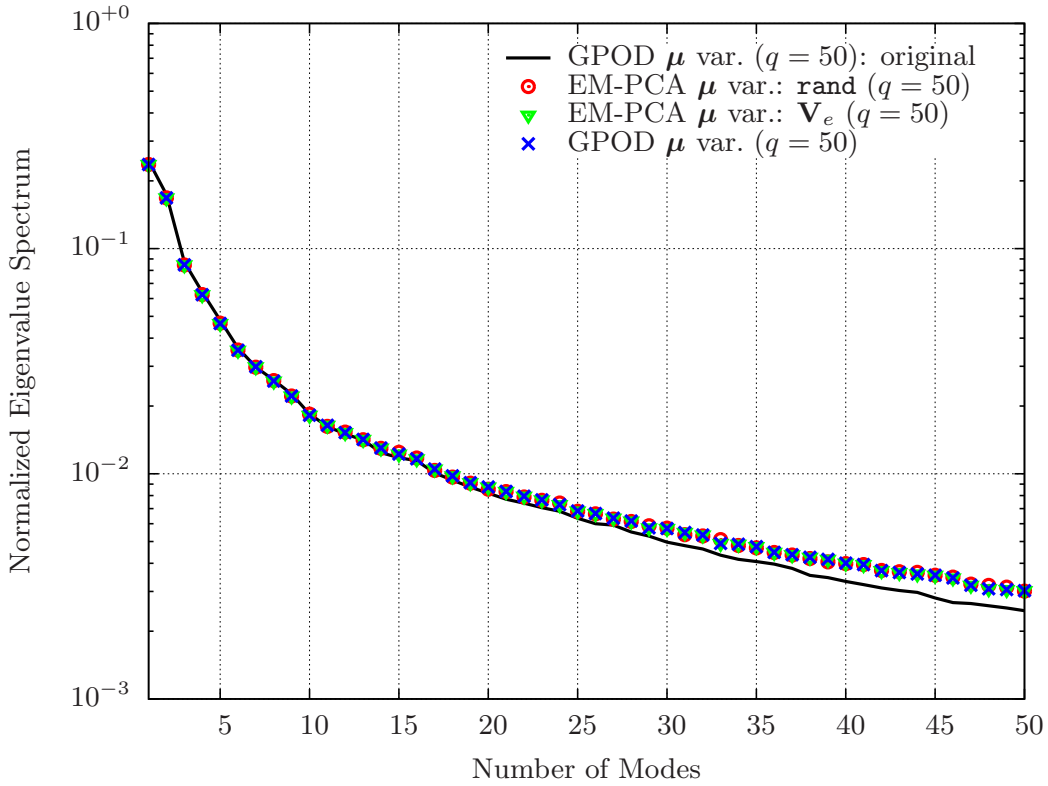


(b) v snapshot ensemble

Figure 67: Eigenspectra of restored u and v velocity components (16.5431% missing)

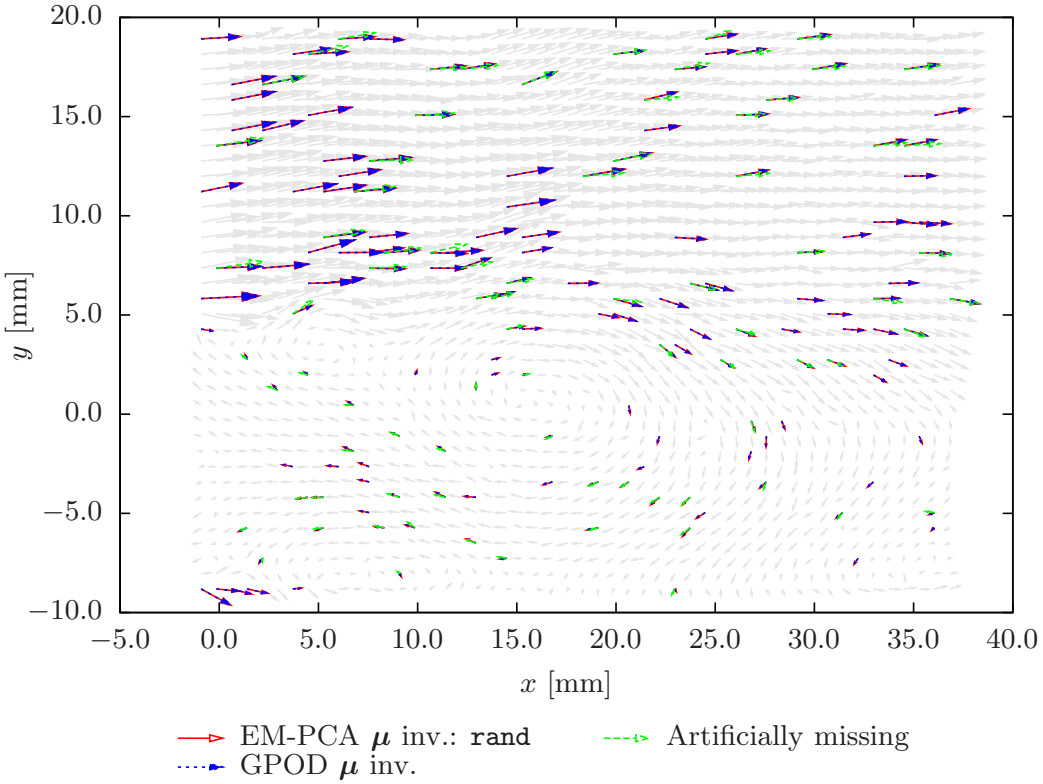


(c) u snapshot ensemble

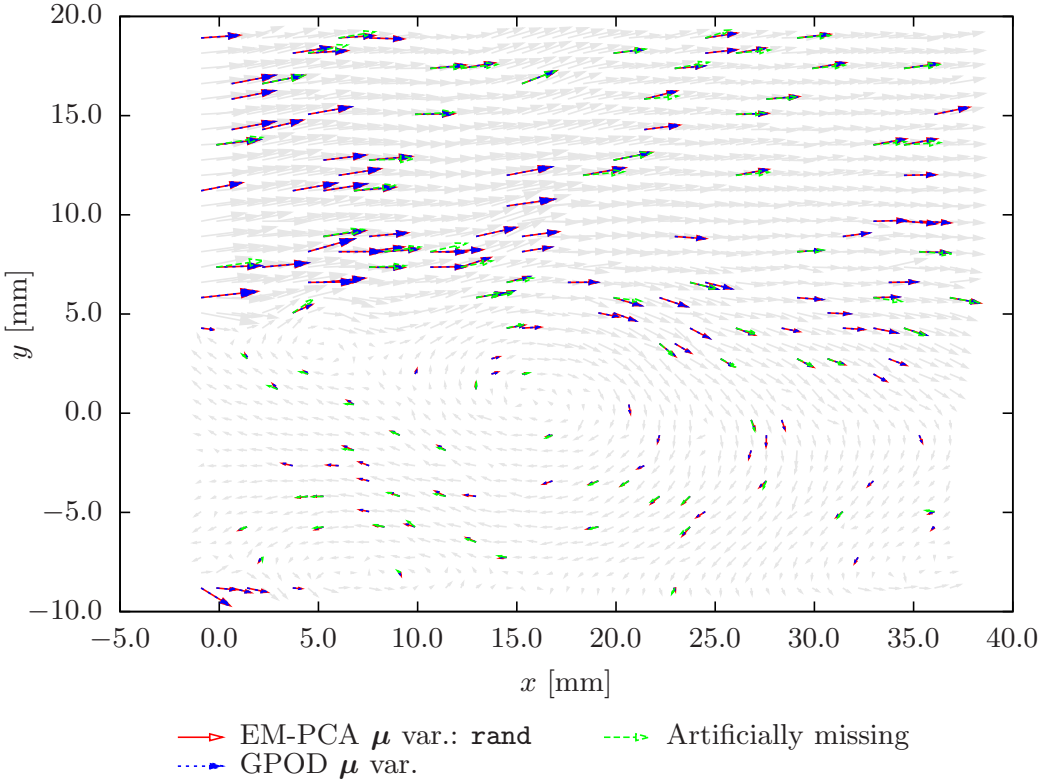


(d) v snapshot ensemble

Figure 67: Eigenspectra of restored u and v velocity components (16.5431% missing)

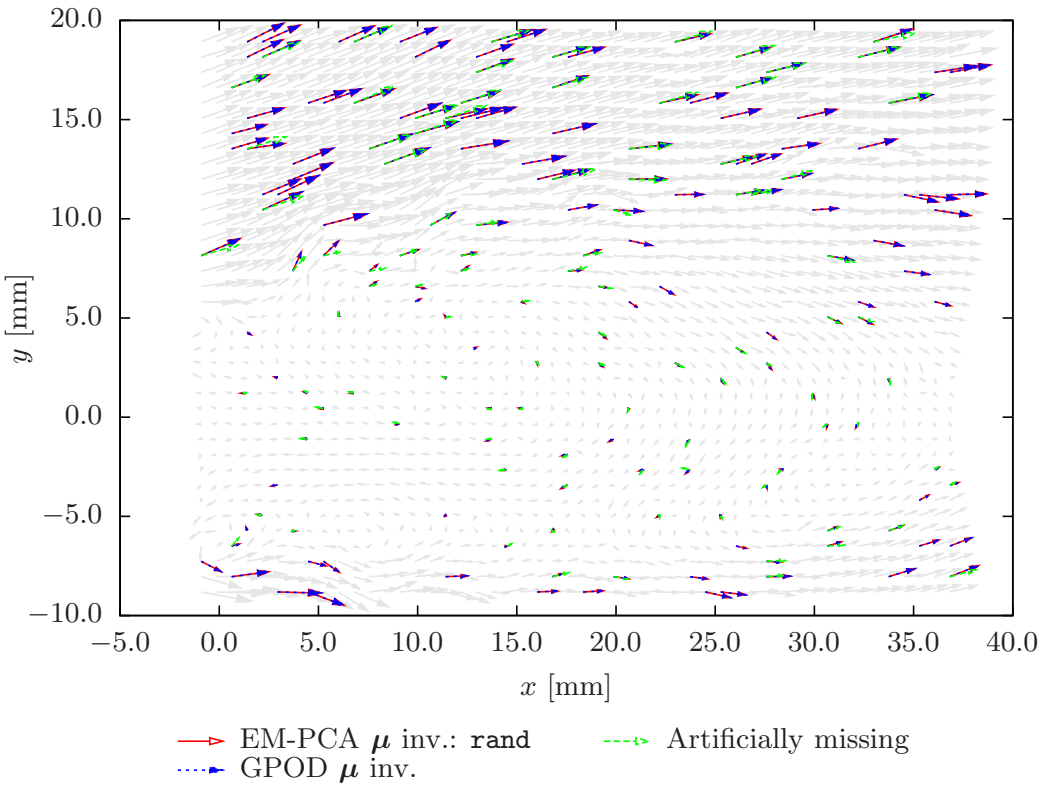


(a) $\boldsymbol{\mu}$ invariant methods

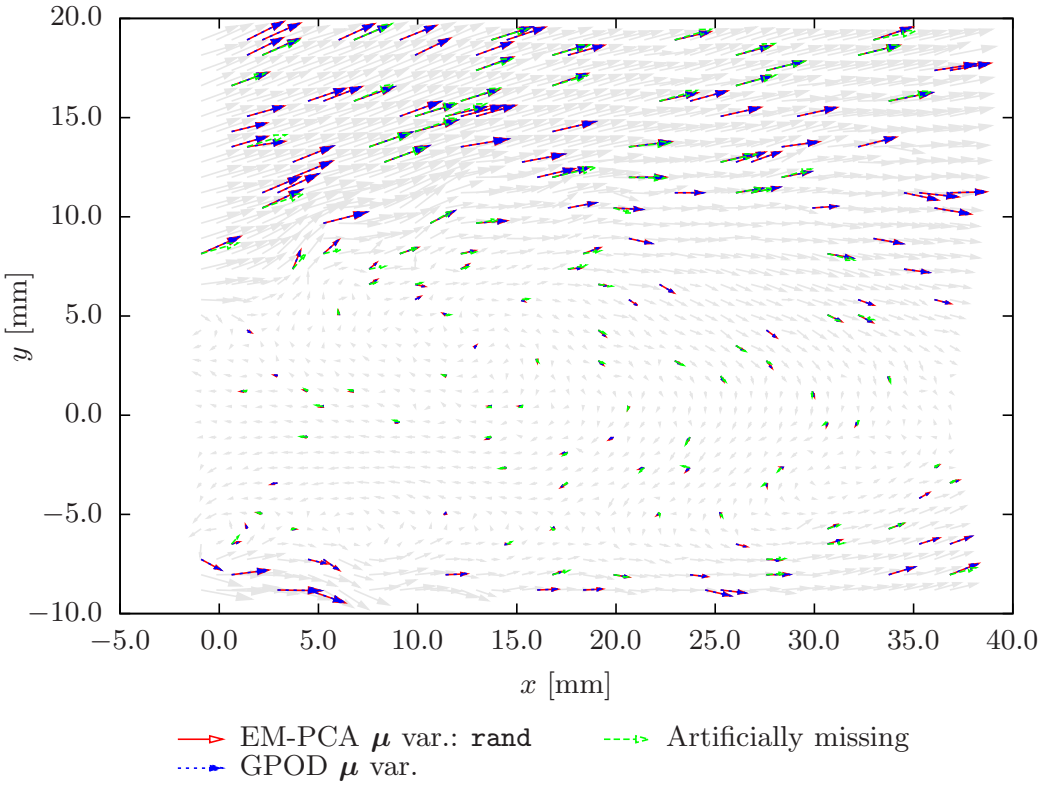


(b) $\boldsymbol{\mu}$ variant methods

Figure 68: Restored 107$^{\text{th}}$ flow velocity snapshot missing 8.7027%: $u(q = 40)$, $v(q = 50)$

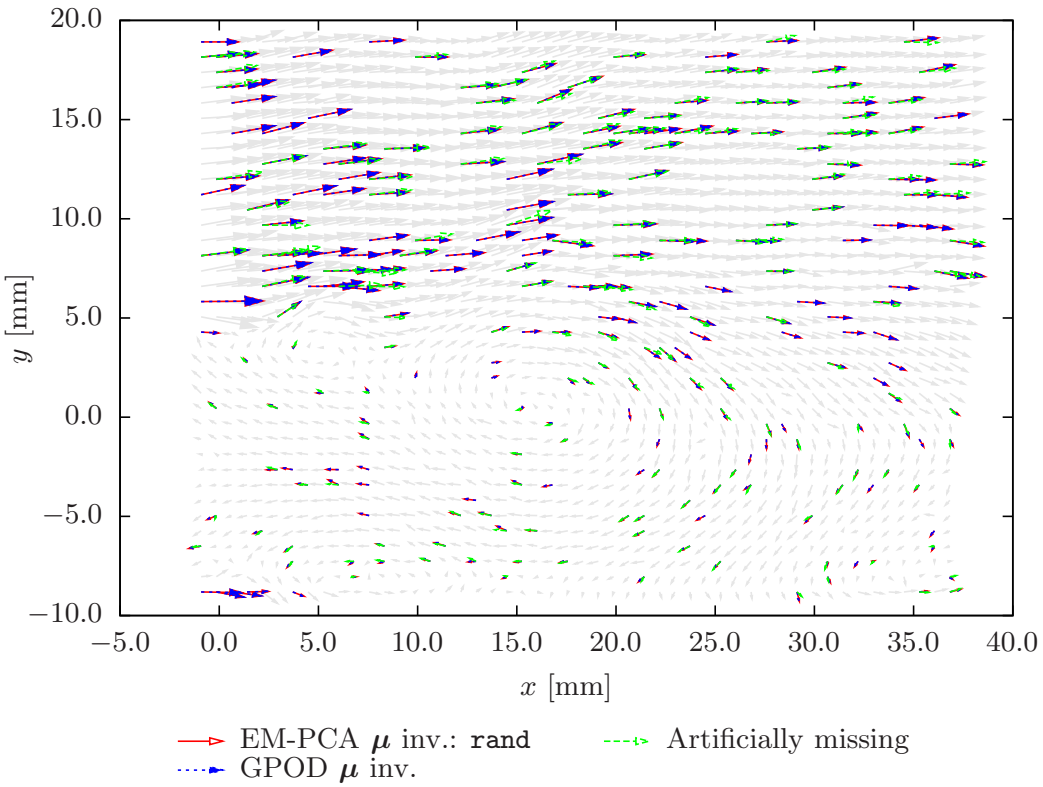


(a) $\boldsymbol{\mu}$ invariant methods

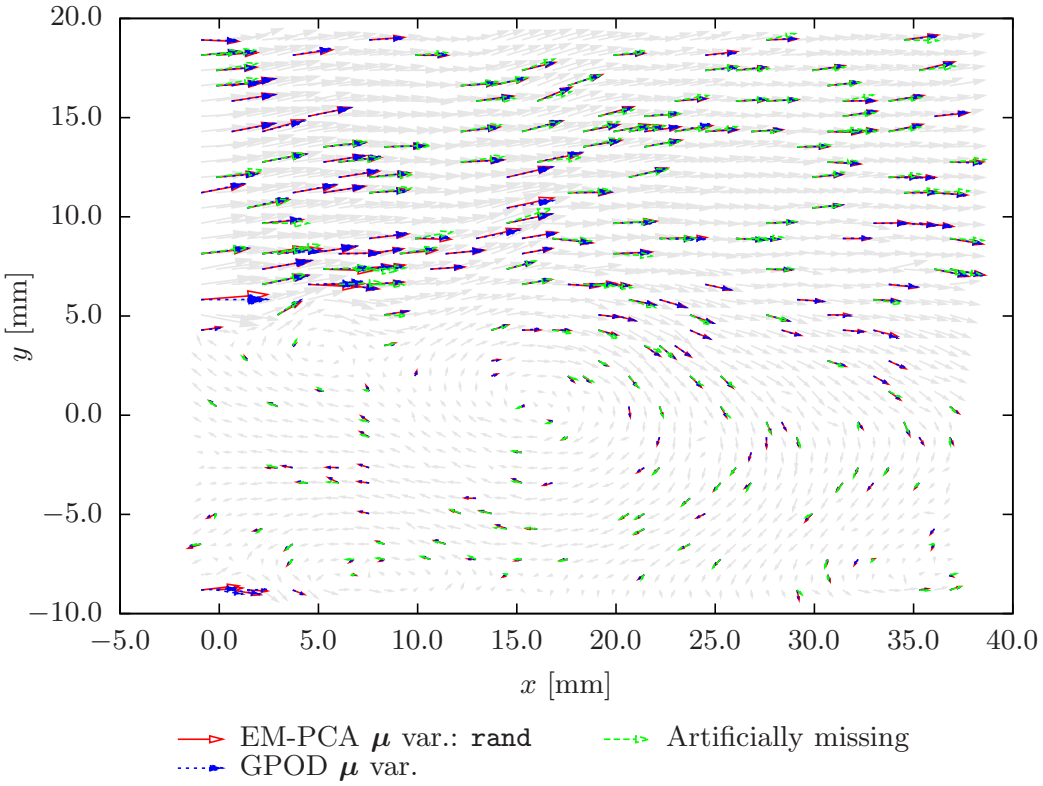


(b) $\boldsymbol{\mu}$ variant methods

Figure 69: Restored 100$^{\text{th}}$ flow velocity snapshot missing 9.4054%: $u(q = 40)$, $v(q = 50)$

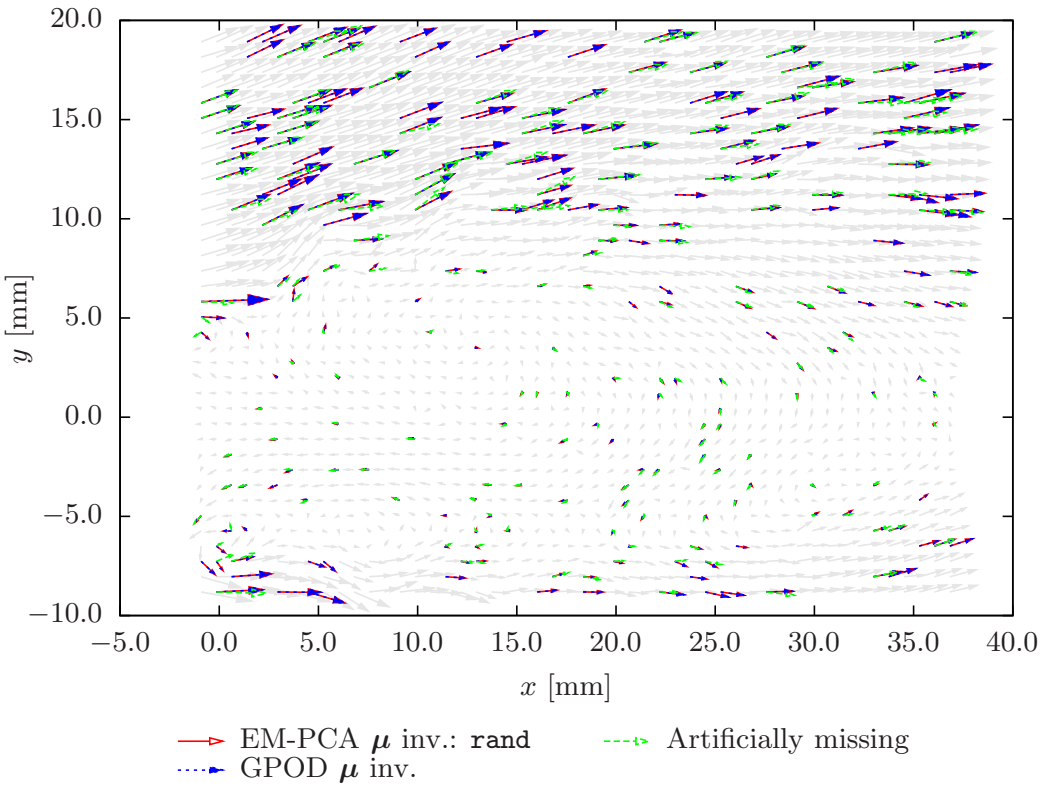


(a) $\boldsymbol{\mu}$ invariant methods

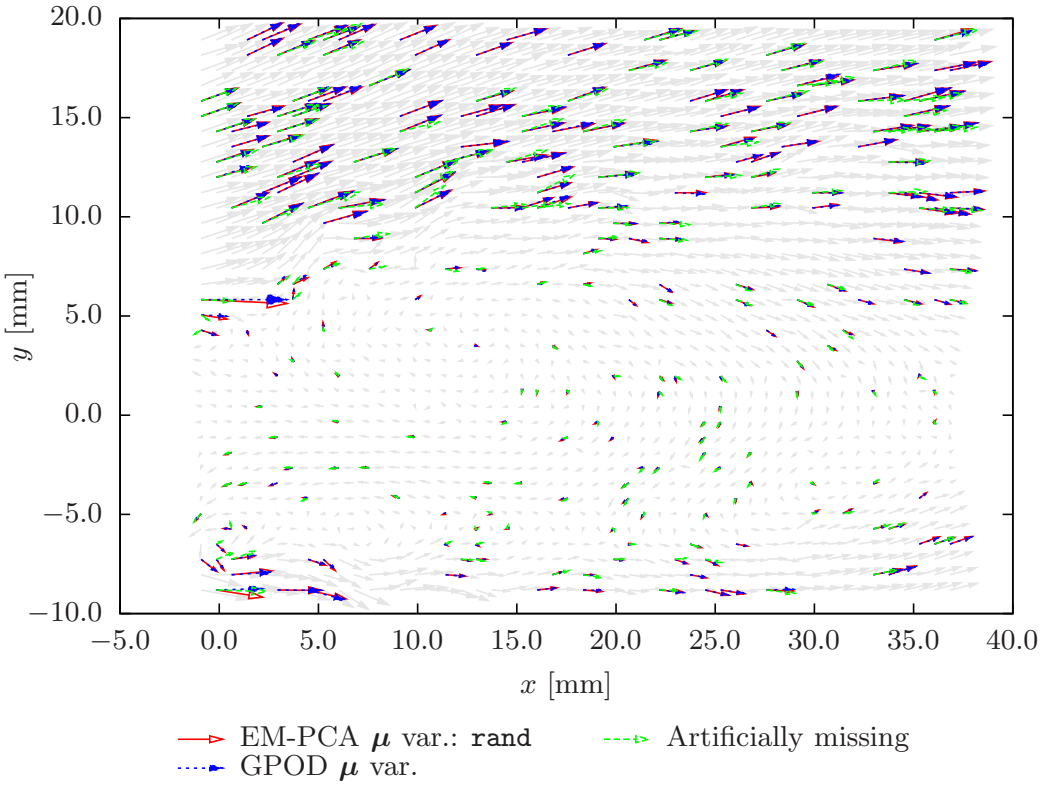


(b) $\boldsymbol{\mu}$ variant methods

Figure 70: Restored 107$^{\text{th}}$ flow velocity snapshot missing 13.7838%: $u(q = 40)$, $v(q = 50)$

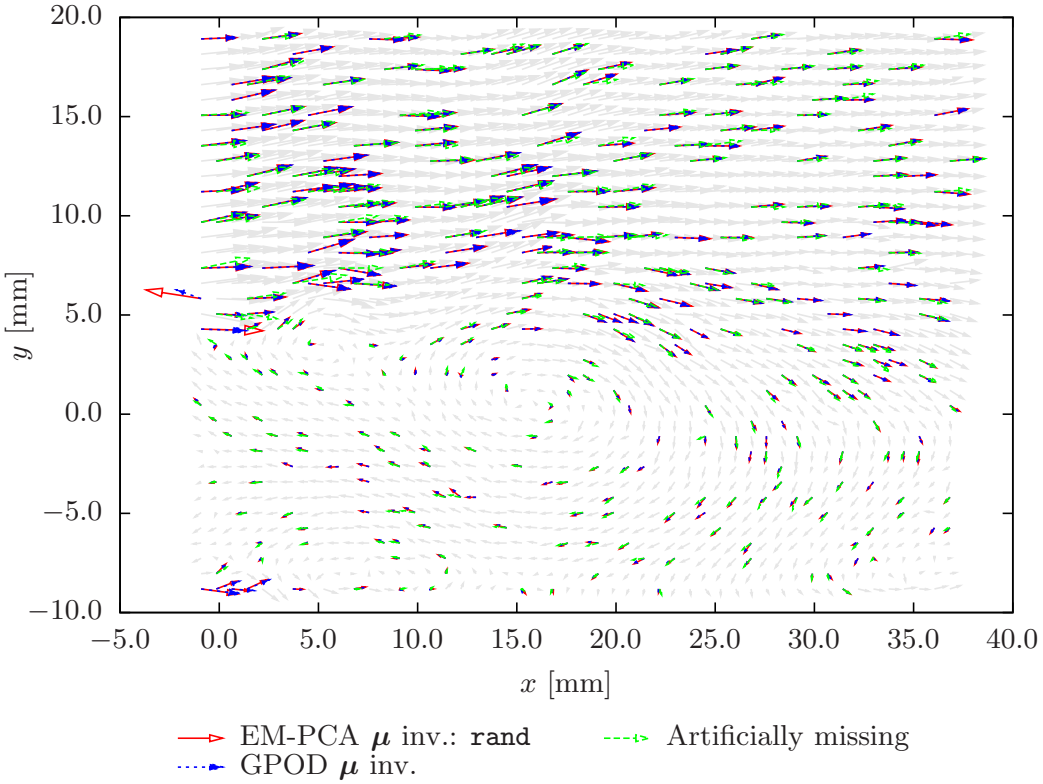


(a) $\boldsymbol{\mu}$ invariant methods

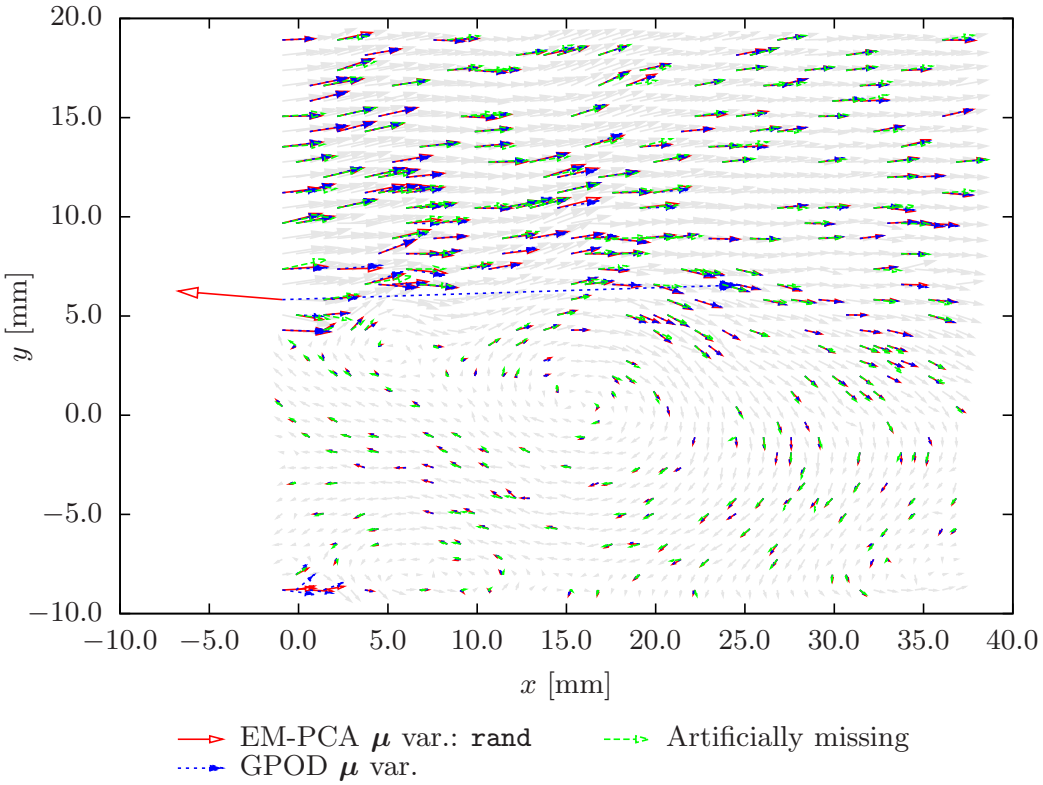


(b) $\boldsymbol{\mu}$ variant methods

Figure 71: Restored 100$^{\text{th}}$ flow velocity snapshot missing 13.8919%: $u(q = 40)$, $v(q = 50)$

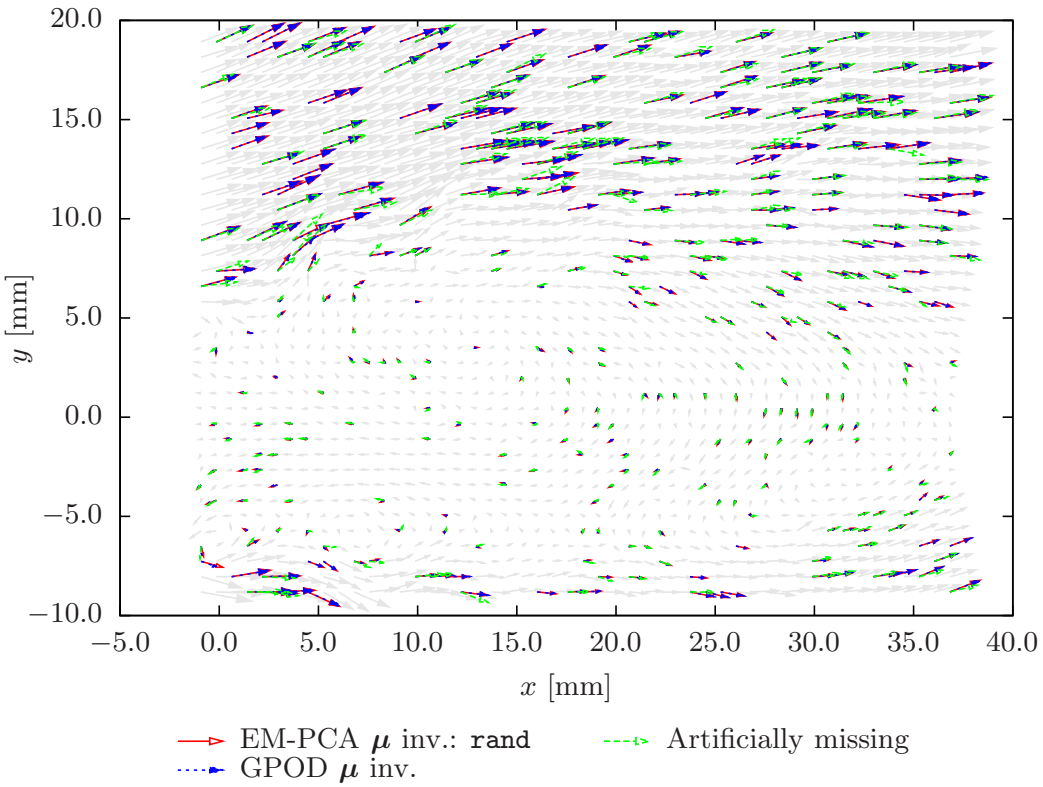


(a) $\boldsymbol{\mu}$ invariant methods

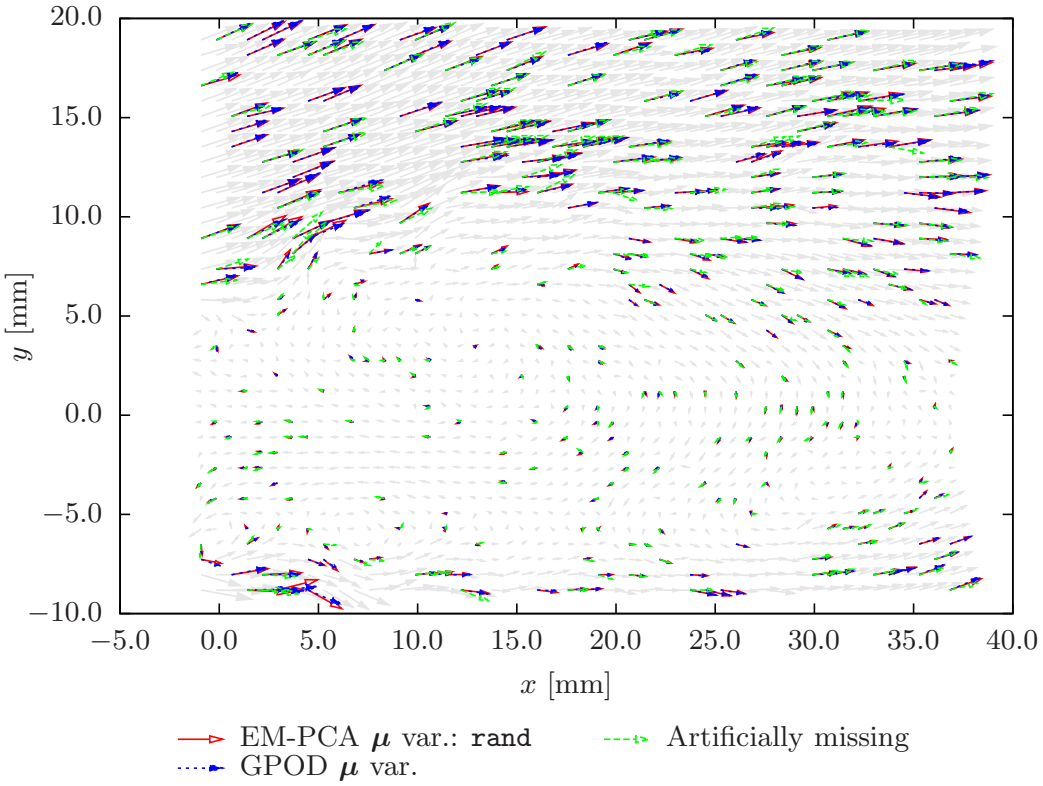


(b) $\boldsymbol{\mu}$ variant methods

Figure 72: Restored 107$^{\text{th}}$ flow velocity snapshot missing 18.7568%: $u(q = 40)$, $v(q = 50)$



(a) $\boldsymbol{\mu}$ invariant methods



(b) $\boldsymbol{\mu}$ variant methods

Figure 73: Restored 100$^{\text{th}}$ flow velocity snapshot missing 18.5946%: $u(q = 40)$, $v(q = 50)$

6.6 Summary

Through comprehensive numerical experiments, this research suggests that the EM-PCA is more efficient than gappy POD for rectifying impaired PIV measurements common in flow experiments. For the sake of PIV data repair, both gappy POD and the EM-PCA are comparable such that they alternate their basis and coefficient evaluations for a least-squares approximation. However, owing to their heterogeneous bases and norms, they end up with distinctive algorithms: gappy POD consists of an orthogonal basis and a weighted least-squares coefficient whereas the EM-PCA comprises a non-orthogonal basis and an ordinary least-squares coefficient. Mainly by virtue of their disparate coefficient evaluations, gappy POD is superior to the EM-PCA with regard to reducing an estimation residual, but it is inferior with regard to its computational effectiveness. In particular, unlike the EM-PCA, gappy POD demands laborious efforts proportional to not only the number of data-missing snapshots but the number of modes. As a result, regarding PIV data reconstruction, the EM-PCA is preferable to gappy POD for the following reasons: (i) PIV data typically contain dispersed spurious measurements, and (ii) complex flow behavior observed in PIV data necessitates a large number of modes due to its high nonlinearity. For this reason, the Lanczos algorithm is not conducive to enhancing gappy POD; the coefficient evaluation is the key impediment in gappy POD, and its advantage quickly vanishes as q increases.

In order to substantiate the efficiency of the EM-PCA over gappy POD, this research performed several tests with the implementations of both methods using a PIV data set from reacting jet-flow experiments. Before the performance investigation, the validation study confirmed that the results of gappy POD fully match those of the EM-PCA in terms of eigenspectra, flow velocity modes, and repaired velocity fields. Afterwards, the first experiment in Section 6.4.1 showed that both steps of the EM-PCA are computationally better than those of gappy POD for a single evaluation. In detail, the coefficient evaluation of gappy POD tends to cause a major performance drawback as the number of modes increases. Next, the second experiment in Section 6.4.2, which compared total times and total iteration numbers, demonstrated that the EM-PCA is more efficient than gappy POD despite its higher iteration numbers. Again, the different coefficient evaluations of both

methods primarily widen their computational time gap as the number of modes increases. Finally, the last two experiments with q increments in Sections 6.4.3 and 6.4.4 exhibited that total time differentials between both methods become conspicuous as q increases, revealing that the EM-PCA is much more effective at a large number of modes.

All in all, the EM-PCA can repair erroneous PIV data as accurate as gappy POD at a lower computational cost, resulting in more computational benefits as the number of modes grows. With regard to future work, a valuable study could be PIV data restoration with the EM-PCA under the effect of measurement noise. Unlike gappy POD, the EM-PCA is able to account for measurement uncertainty with an isotropic Gaussian distribution through its intrinsic error model. This future work may overcome the limitation of gappy POD in addressing experimental uncertainty for PIV data restoration.[57,58]

CHAPTER VII

CONCLUSIONS AND FUTURE WORK

7.1 Concluding Remarks

Motivated by the flexible applicability of the EM-PCA for not only intact but also gappy data, this research thoroughly scrutinized the EM-PCA, comparing it to both POD and gappy POD. For this purpose, the advantages and disadvantages of the EM-PCA were explored and compared to those of POD first and gappy POD next. With regard to the first comparative study of the EM-PCA and POD, the factor-loading matrix of the PPCA, i.e., one of the PPCA parameters, is known as analytically germane to a POD basis. As opposed to an orthogonal POD basis, an EM-PCA basis is non-orthogonal for its immanent scaling and rotation operations. However, those extra linear operations can be simply removed through orthogonalization, resulting in an orthogonal basis derived from a non-orthogonal basis. Subsequently, the numerical validation results of the EM-PCA and POD obtained with FPE and Euler CFD simulations confirmed that the EM-PCA generates the same eigenspectra and basis vectors as gappy POD. Despite identical validation results, the EM-PCA turned out to be less efficient than POD mainly because of its poor convergence behavior. The slow convergence of the EM-PCA indicates that the EM-PCA is prone to suffering from hard-to-decaying, low-frequency errors as it marches through iterations.

As mentioned above, the theoretical relationship between the EM-PCA and POD is clear, and yet that between the EM-PCA and gappy POD is obscure. Owing to their disparate formulation approaches, to estimate missing data, the EM-PCA depends on an observation probability model whereas gappy POD relies on a least-squares formulation. In order to effectively juxtapose both the EM-PCA and gappy POD for a comparative study, this research reformulated gappy POD and reinterpreted the EM-PCA from the unifying least-squares perspective. As a result, the unifying perspective revealed that both

the EM-PCA and gappy POD address similar least-squares problems; however, their least-squares problems comprise dissimilar bases and norms. In detail, the EM-PCA utilizes a non-orthogonal basis and the L^2 norm whereas gappy POD employs an orthogonal basis and the gappy norm. Furthermore, this research delved into the effects of the different bases and norms, which eventually predetermine the theoretical and numerical traits of the EM-PCA and gappy POD. First, the theoretical analysis of the different basis and norm effects was inconclusive for discerning which basis or norm is superior at reducing estimation residuals. For instance, a POD basis evaluated in iterations for gappy POD is an estimate of the unknown true POD basis; thus, it does not possess the desirable property of a minimal projection error for the true intact data. Likewise, a norm pertains to a curve-fitting method such that the L^2 and gappy norms conceptually perform regression and interpolation, respectively; however, their advantage to the other is indeterminate.

In order to complement the previous insufficient analytical approach, this research measured the effects of the different bases and norms in terms of an RMSE in missing data estimation. In particular, for this quantitative investigation, two "in-between" algorithms of gappy POD and the EM-PCA were developed through a combination of their bases and norms. These hybrid algorithms were devised such that only a basis or a norm difference bridges the gap between the hybrids and the originals; thus, their comparisons can isolate each basis and norm difference effect. The numerical quantification with RMSEs generated two artificially marred sample data sets: the first sample data set contained missing data confined to only a single snapshot, and the second sample data set held missing data spread over all the snapshots. With the two sample data sets, both basis and norm effects were effectively assessed through the RMSE comparisons obtained by the original and the hybrid algorithms. In addition, the numerical efficiency of all the algorithms were examined in terms of total iteration numbers along with total computational time decomposed into their basis and coefficient evaluations. The computational efficiency tests showed that a missing data structure affects the performance of the reconstruction method; gappy POD outperforms the EM-PCA for the first sample data set, and vice versa for the second sample data set.

According to the results of different basis and norm effects accessed with an RMSE, a norm turned out to have a more considerable impact on missing data estimation than a basis. Beginning with the basis effect, in the first sample data set, the basis difference produced a little RMSE differentials, but it generated much lower RMSE differentials than the norm difference. In the second sample data set, the basis difference resulted in a virtually insignificant effect because an approximate orthogonal basis was almost equally as good as an approximate non-orthogonal basis. In particular, RMSE comparisons showed that an approximate non-orthogonal basis randomly initialized could reduce RMSE even more than an approximate orthogonal basis. Regarding the norm effect, the gappy norm turned out to be more effective at reducing estimation errors than the L^2 norm, and led to considerably fewer iterations. Despite its superior convergence capability, the gappy norm demanded more computational effort for coefficient evaluations than the L^2 norm because it required as many projection evaluations as the number of data-missing snapshots. These numerical characteristics entailed by the norms explain why gappy POD outperformed the EM-PCA for the first sample data set, whose number of data-missing snapshots was one, but it did not for the second sample data set, whose number of data-missing snapshots was more than one.

After the theoretical and numerical examination of the EM-PCA and the POD methods, this research identified the benefits of the EM-PCA for two applications: basis extraction and missing data estimation with an incomplete data set. For the first application, this research capitalized on the EM-PCA to construct a POD-based ROM of NPSS because some NPSS results are typically absent due to failed analyses. Since POD fails to deal with a gappy NPSS snapshot ensemble, the EM-PCA was employed for a POD basis evaluation. In conjunction with the EM-PCA, this research exploited neural networks to effectively explore a coefficient space to predict the behavior of NPSS in an unseen input parameter set. Although a computational efficiency investigation is not of primary interest, this research also showed that the EM-PCA generated a POD basis faster than gappy POD due to multiple data-missing snapshots in an NPSS snapshot ensemble. For the second application, the EM-PCA was utilized for rectifying impaired PIV measurements common

in flow experiments using the PIV technique. The validation study confirmed that both the EM-PCA and gappy POD yield identical eigenspectra, flow velocity modes, and repaired velocity fields. However, since every snapshot of the PIV data set contained unreliable measurements, the EM-PCA is computationally preferable to gappy POD for spurious PIV data correction.

In conclusion, based on the comprehensive investigations through various numerical experiments, this research noticed the following: (i) for an intact data set, the EM-PCA is less efficient than POD for basis extraction because of its poor convergence behavior inherited from the EM algorithm, and (ii) for a gappy data set, the EM-PCA is more efficient than gappy POD for basis extraction and missing data estimation by virtue of its L^2 norm insofar as the data set includes multiple data-missing snapshots. Note that the relative computational efficiency of the EM-PCA with respect to that of gappy POD crucially hinges on the number of data-missing snapshots. Figure 74 depicts the two typical missing data types of missing data estimation applications. For instance, Figure 74(a)

$$\begin{bmatrix} x & x & x & \cdots & * & x & x \\ x & x & x & \cdots & * & x & x \\ x & x & x & \cdots & * & x & x \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ x & x & x & \cdots & x & x & x \\ x & x & x & \cdots & x & x & x \\ x & x & x & \cdots & x & x & x \end{bmatrix} \qquad \begin{bmatrix} x & * & x & \cdots & x & x & x \\ x & x & * & \cdots & * & x & x \\ x & x & x & \cdots & x & x & * \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ x & x & x & \cdots & x & * & x \\ * & x & x & \cdots & * & x & x \\ x & x & * & \cdots & x & x & * \end{bmatrix}$$

(a) Missing data only at a single snapshot (b) Missing data across all the snapshots

Figure 74: Typical missing data structures

represents the missing data structure of such applications as experimental and numerical flow data assimilation, inverse airfoil design, and so forth. Likewise, Figure 74(b) indicates a missing data type used for the following applications: basis extraction from the results of NPSS analysis and PIV data restoration in Chapters 5 and 6, respectively. Since the first missing data type in Figure 74(a) has only one data-missing snapshot, gappy POD

outperforms the EM-PCA. Other than that, as an incomplete data set accumulates more and more data-missing snapshots, the EM-PCA is expected to be more efficient than gappy POD. Therefore, the EM-PCA is recommended for applications whose missing data are spread over an entire data set, as shown in Figure 74(b).

7.2 Research Questions and Hypotheses Revisited

With the results of the exhaustive comparative studies presented in this thesis, this section revisits the previously formulated research questions and concomitant hypotheses. To achieve Research Objective 1, this thesis addressed the following Research Questions 1.1 to 1.3 to verify Hypotheses 1.1 to 1.3.

Research Question 1.1 For an intact data set, is the EM-PCA computationally competitive with POD methods for basis extraction?

Answer No, it is not. The computational performance investigation in Chapter 3 showed that the EM-PCA can outperform snapshot POD only when the number of modes to extract is one. As the number of modes to extract escalates, the EM-PCA did not surpass gappy POD.

Research Question 1.2 For an incomplete data set whose missing data are only at a single snapshot, is the EM-PCA computationally competitive with gappy POD for basis extraction and missing data estimation?

Answer No, it is not. The computational performance investigation in Chapter 4 showed that the EM-PCA required more computational time than gappy POD for an incomplete data set whose missing data are only at a single snapshot.

Research Question 1.3 For an incomplete data set whose missing data are across all the snapshots, is the EM-PCA computationally competitive with gappy POD for basis extraction and missing data estimation?

Answer Yes, it is. The computational performance investigation in Chapter 4 showed that the EM-PCA required less computational time than gappy POD for an incomplete data set whose missing data are across all the snapshots.

Based on the answers to Research Questions 1.1 to 1.3, their corresponding hypotheses are evaluated as follows.

Hypothesis 1.1 For an intact data set, the EM-PCA takes less computational time than POD.

Verification For an intact data set, the EM-PCA is less efficient than POD. Therefore, Hypothesis 1.1 is rejected.

Hypothesis 1.2 For an incomplete data set whose missing data are only at a single snapshot, the EM-PCA takes less computational time than gappy POD.

Verification For an incomplete data set whose missing data are only at a single snapshot, the EM-PCA is less efficient than gappy POD. Therefore, Hypothesis 1.2 is rejected.

Hypothesis 1.3 For an incomplete data set whose missing data are across all the snapshots, the EM-PCA takes less computational time than gappy POD.

Verification For an incomplete data set whose missing data are across all the snapshots, the EM-PCA is more efficient than gappy POD. Therefore, Hypothesis 1.3 is accepted.

Through the verifications of Hypotheses 1.1 to 1.3, Methodological Hypothesis 1, shown in the below, is partially accepted such that the EM-PCA is computationally more efficient than gappy POD for an incomplete data set whose missing data are across all the snapshots.

Methodological Hypothesis 1 The EM-PCA yields identical results to those of POD and gappy POD, but it is computationally more efficient than POD and gappy POD in terms of computational time.

After all, Research Objective 1, in the below, is accomplished in this thesis.

Research Objective 1 To facilitate the use of the EM-PCA in addressing the problems of aerospace engineering, this research attempts to theoretically and numerically compare and contrast the EM-PCA and to both POD and gappy POD for basis extraction and missing data estimation.

While this research has strived to address Research Questions 1.2 and 1.3, the relationship between the EM-PCA and gappy POD was nebulous, necessitating further investigation. Thus, this research set the second research objective in Chapter 1 as follows.

Research Objective 2 To compare and contrast the EM-PCA to gappy POD, this research attempts to identify the formulation similarities and disparities of the EM-PCA and gappy POD.

After this thesis manipulated the equations of the EM-PCA and gappy POD in Chapter 4, it was able to reveal that both methods pertain to least-squares formulations. Based on their formulation similarity, Methodological Hypothesis 2, in the below, was established and verified in Chapter 4 in an effort to achieve Research Objective 2.

Methodological Hypothesis 2 A unifying least-squares perspective integrates both the EM-PCA and gappy POD within a common formulation framework.

With the help of the unifying least-squares perspective, a basis and a norm are found to be two crucial factors that differentiate the EM-PCA and gappy POD, which raised Research Question 2.1 in Chapter 4.

Research Question 2.1 What are the effects of the disparate bases and norms on estimation error reduction and the computational performance of the EM-PCA and gappy POD?

Through systematic comparative studies in Chapter 4, Research Question 2.1 was addressed as follows.

Answer The norm difference of the EM-PCA and gappy POD determines their estimation residual reductions. For instance, due to the gappy norm, gappy POD is superior at reducing estimation residuals, but inferior at computational performance. Likewise, owing to the L^2 norm, the EM-PCA is inferior at reducing estimation residuals, but superior at computational performance.

Based on the observations in Chapter 4, Hypothesis 2.1 was formulated.

Hypothesis 2.1 A norm selection affects estimation error reduction more than a basis selection.

In order to verify Hypothesis 2.1, this thesis measures the computational performance of the original and hybrid algorithms with the PIV data, as shown in Figures 75 and 76, and with the gross thrust data set, as shown in Figure 77. Similar to the comparative discussion in Chapter 4, the comparison of the gappy POD and Hybrid 2 reveals the norm difference under the same $\mathbf{V}_q$ basis; likewise, the comparison of the EM-PCA and Hybrid 2 shows the basis difference under the identical L^2 norm. Overall, Figures 75 to 77 are helpful for evaluating Hypothesis 2.1 as follows.

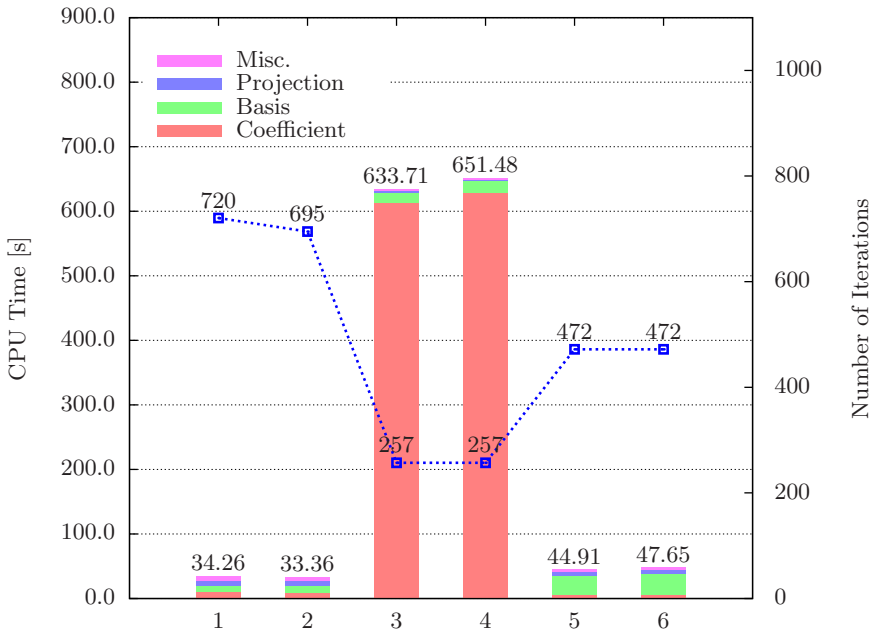
Verification Hypothesis 2.1 is verified for the incomplete data set containing missing data across all the snapshots, such as the u and v snapshot ensembles of the PIV data sets and the gross thrust data set of NPSS.

Certainly, the results in Figures 75 to 77 are not sufficient, and more substantiating results of comparative studies are required to thoroughly verify Hypothesis 2.1.

From a computational time aspect, both Figures 75 and 76 show that the norm difference yields more differentials in computational time than the basis difference when q is large such that $q = 40$. By contrast, Figure 77 shows that the basis difference produces more differentials in computational time than the norm difference because of the large number of snapshots such as 500. Note that the effects of the disparate bases and norms cannot be evaluated with RMSE histories since no true intact data are available, unlike the artificially manipulated flow simulation data in Chapter 4.

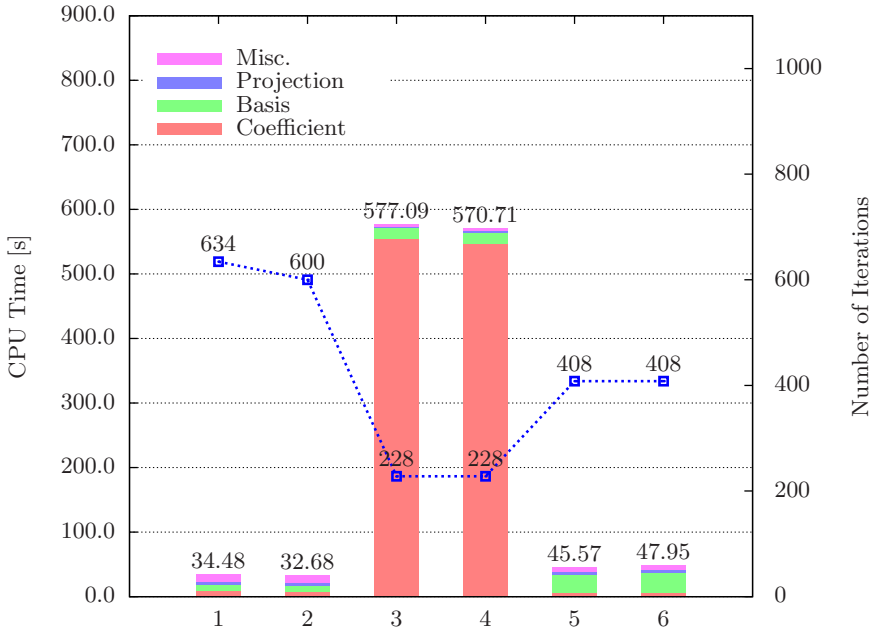
7.3 Recommendations for Future Work

In theory, the EM-PCA estimates missing observations based on the Gaussian probability model of PPCA. Likewise, gappy POD also implicitly hinges on a Gaussian probability model in the sense that it relies on two parameters, mean and covariance, for missing data estimation. However, purely from a probabilistic standpoint, the Gaussian probability assumption is too simple to sufficiently delineate complex high-fidelity flow simulation data.



1: EM-PCA $\boldsymbol{\mu}$ inv.: `rand` ($q = 40$)
2: EM-PCA $\boldsymbol{\mu}$ inv.: $\mathbf{V}_e$ ($q = 40$)
3: GPOD $\boldsymbol{\mu}$ inv. ($q = 40$)
4: GPOD $\boldsymbol{\mu}$ inv.: Lanczos ($q = 40$)
5: Hybrid2 $\boldsymbol{\mu}$ inv. ($q = 40$)
6: Hybrid2 $\boldsymbol{\mu}$ inv.: Lanczos ($q = 40$)

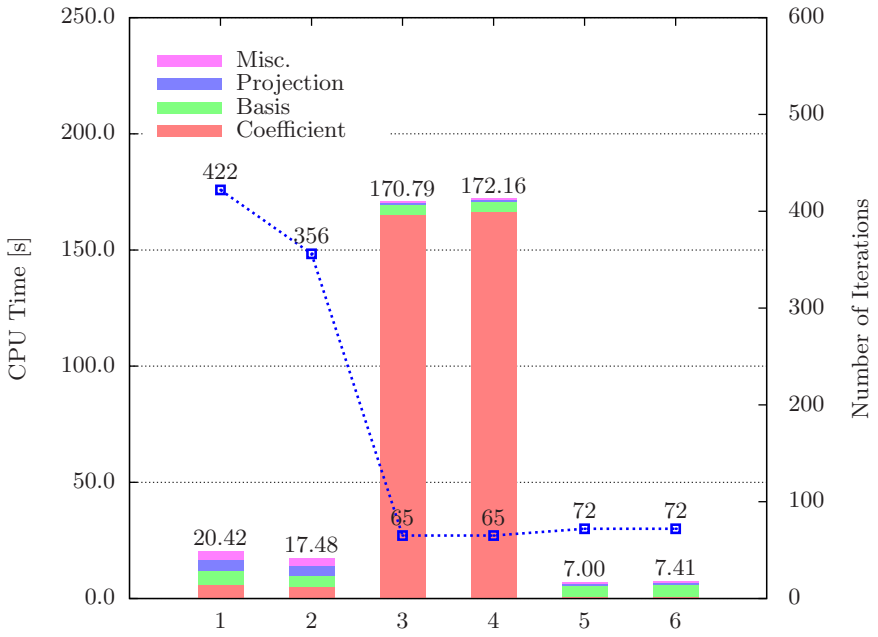
(a) $\boldsymbol{\mu}$ invariant methods



1: EM-PCA $\boldsymbol{\mu}$ var.: `rand` ($q = 40$)
2: EM-PCA $\boldsymbol{\mu}$ var.: $\mathbf{V}_e$ ($q = 40$)
3: GPOD $\boldsymbol{\mu}$ var. ($q = 40$)
4: GPOD $\boldsymbol{\mu}$ var.: Lanczos ($q = 40$)
5: Hybrid2 $\boldsymbol{\mu}$ var. ($q = 40$)
6: Hybrid2 $\boldsymbol{\mu}$ var.: Lanczos ($q = 40$)

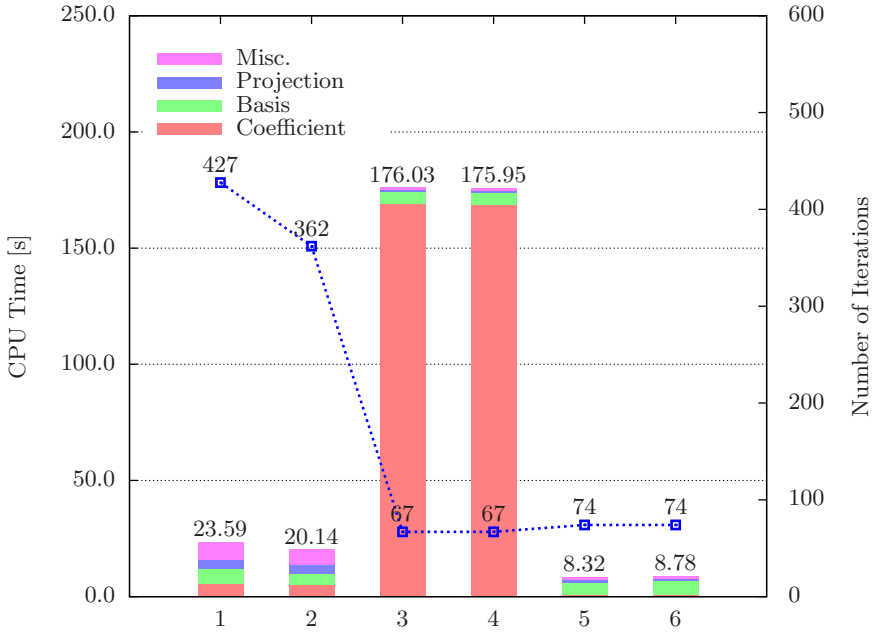
(b) $\boldsymbol{\mu}$ variant methods

Figure 75: Computational time and the number of iterations of the u snapshot ensemble



1: EM-PCA $\boldsymbol{\mu}$ inv.: `rand` ($q = 40$)
2: EM-PCA $\boldsymbol{\mu}$ inv.: $\mathbf{V}_e$ ($q = 40$)
3: GPOD $\boldsymbol{\mu}$ inv. ($q = 40$)
4: GPOD $\boldsymbol{\mu}$ inv.: Lanczos ($q = 40$)
5: Hybrid2 $\boldsymbol{\mu}$ inv. ($q = 40$)
6: Hybrid2 $\boldsymbol{\mu}$ inv.: Lanczos ($q = 40$)

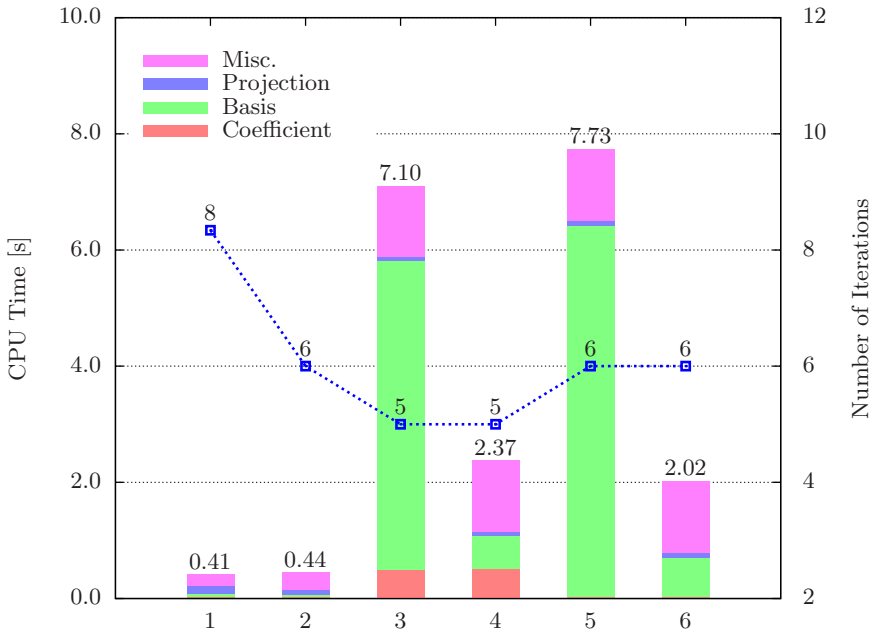
(a) $\boldsymbol{\mu}$ invariant methods



1: EM-PCA $\boldsymbol{\mu}$ var.: `rand` ($q = 40$)
2: EM-PCA $\boldsymbol{\mu}$ var.: $\mathbf{V}_e$ ($q = 40$)
3: GPOD $\boldsymbol{\mu}$ var. ($q = 40$)
4: GPOD $\boldsymbol{\mu}$ var.: Lanczos ($q = 40$)
5: Hybrid2 $\boldsymbol{\mu}$ var. ($q = 40$)
6: Hybrid2 $\boldsymbol{\mu}$ var.: Lanczos ($q = 40$)

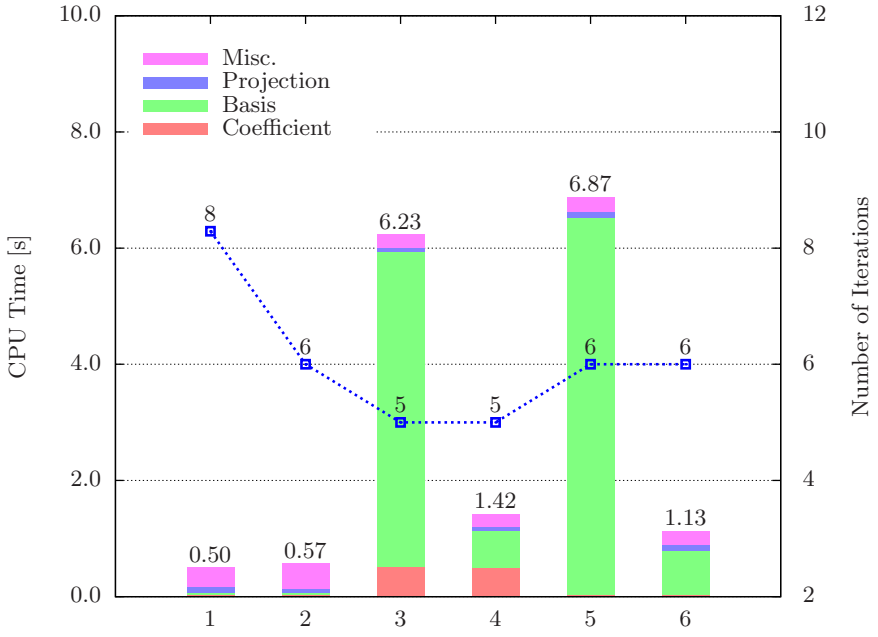
(b) $\boldsymbol{\mu}$ variant methods

Figure 76: Computational time and the number of iterations of the v snapshot ensemble



1: EM-PCA $\boldsymbol{\mu}$ inv.: `rand` ($q = 4$)
2: EM-PCA $\boldsymbol{\mu}$ inv.: $\mathbf{V}_e$ ($q = 4$)
3: GPOD $\boldsymbol{\mu}$ inv. ($q = 4$)
4: GPOD $\boldsymbol{\mu}$ inv.: Lanczos ($q = 4$)
5: Hybrid2 $\boldsymbol{\mu}$ inv. ($q = 4$)
6: Hybrid2 $\boldsymbol{\mu}$ inv.: Lanczos ($q = 4$)

(a) $\boldsymbol{\mu}$ invariant methods



1: EM-PCA $\boldsymbol{\mu}$ var.: `rand` ($q = 4$)
2: EM-PCA $\boldsymbol{\mu}$ var.: $\mathbf{V}_e$ ($q = 4$)
3: GPOD $\boldsymbol{\mu}$ var. ($q = 4$)
4: GPOD $\boldsymbol{\mu}$ var.: Lanczos ($q = 4$)
5: Hybrid2 $\boldsymbol{\mu}$ var. ($q = 4$)
6: Hybrid2 $\boldsymbol{\mu}$ var.: Lanczos ($q = 4$)

(b) $\boldsymbol{\mu}$ variant methods

Figure 77: Computational time and the number of iterations of gross thrust

To verify the Gaussianity of observations, one can evaluate the kurtosis* of observations and compare it with that of a Gaussian distribution, which is three. For instance, the kurtosis of the C_p snapshot ensemble with an Euler CFD solver in Chapter 4 is 7.25592, which is much higher than three. Apparently, from a probabilistic point of view, the Gaussian model assumption is improper for the Euler CFD simulation data. Therefore, other complicated probability models using more than two parameters would be desirable in a probabilistic sense as long as the probability model fully represented a sample data set.† For this purpose, one could benefit from other probabilistic generalizations of PCA such as the exponential-type probability distributions for PCA[51] or a student t-distribution for PCA.[22]

Another interesting future study is to delineate the complete theoretical interrelationship between probabilistic and deterministic POD formulations. As PPCA is probabilistically relevant to the standard POD, so is dual probabilistic principal component analysis (DPPCA)[29] to the snapshot POD. Lawrence conceived DPPCA by adopting a Bayesian perspective for PPCA in order to admit an inner product in the formulation of DPPCA. Through an inner product that can be replaced with a nonlinear kernel function, DPPCA can employ kernel methods to nonlinearly expand PPCA. As a result of the Bayesian interpretation, instead of marginalizing $\mathbf{X}$ and optimizing $\mathbf{W}$ as in PPCA, DPPCA marginalizes $\mathbf{W}$, treating it as a random variable, and optimizes $\mathbf{X}$, treating it as a probability parameter. After all, the formulation of DPPCA is identical to that of PPCA except that the roles of $\mathbf{W}$ and $\mathbf{X}$ are reversed. Conceptually, DPPCA is equivalent to applying PPCA to a transposed snapshot ensemble $\mathbf{Y}^{\mathrm{T}}$ as is the snapshot POD, which carries out the standard POD procedure on $\mathbf{Y}^{\mathrm{T}}$. As an illustration, Figure 78 provides an overall view of the theoretical interrelationship between deterministic and probabilistic POD formulations.

From a numerical performance aspect, another interesting future study is to investigate convergence acceleration methods for the EM-PCA. The observed characteristic of poor

*A kurtosis of less than three implies that the distribution of observations has a lower, wider peak around its mean and thinner tails, so it is less outlier-prone than a Gaussian distribution. On the other hand, a kurtosis of more than three indicates that the distribution of observations has a more acute, narrower peak around its mean and flatter tails, so it is more outlier-prone than a Gaussian distribution.

†Note that a Gaussian distribution is the simplest probability model, requiring only two parameters: mean and variance.

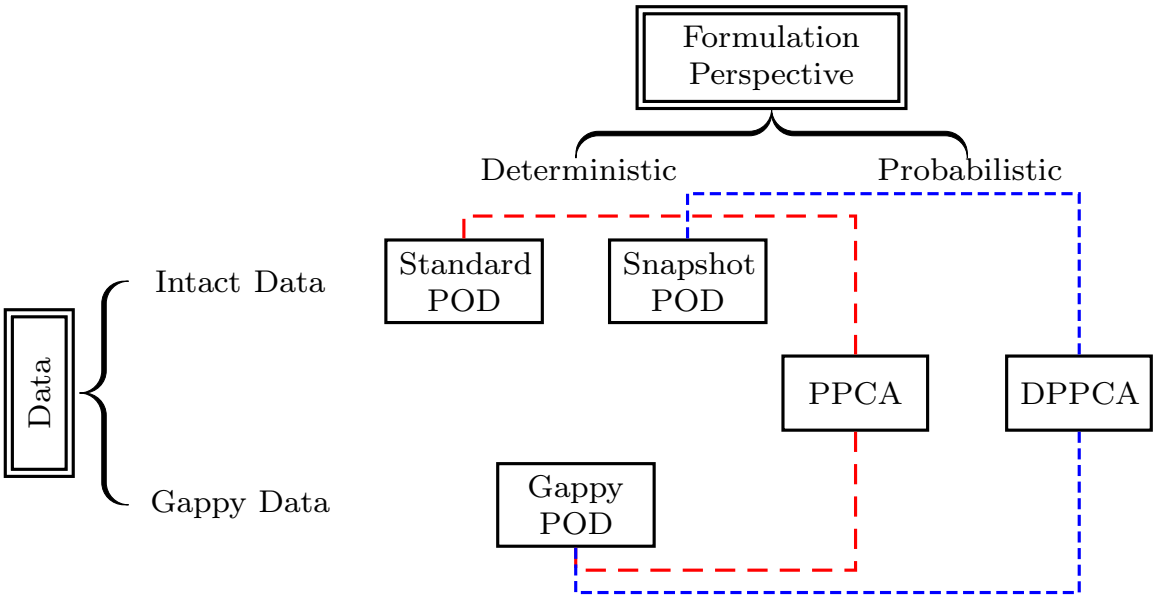
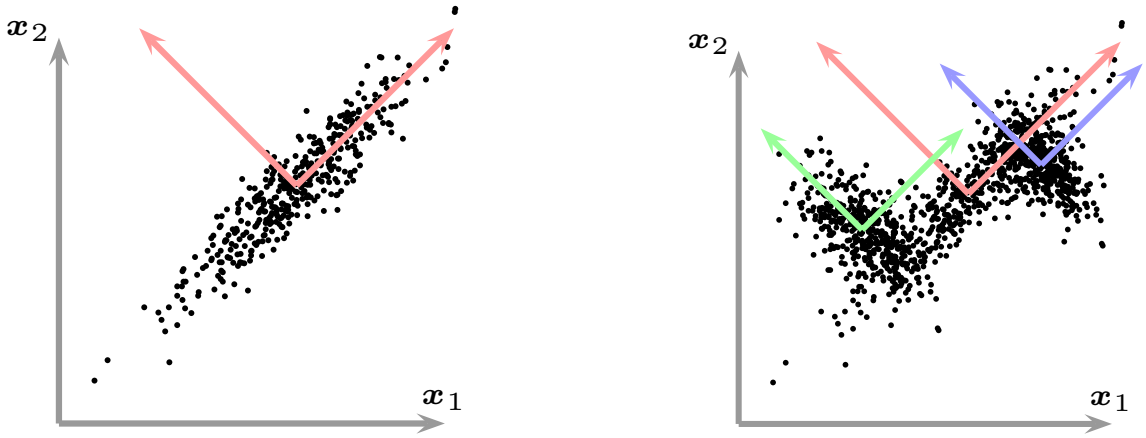


Figure 78: Interrelationship between deterministic and probabilistic POD methods

convergence of the EM-PCA in Chapter 3 mainly originates from the slow convergence of the EM algorithm.[59,60] Among several convergence acceleration techniques, overrelaxation is a simple yet powerful technique that allows an iterative method to take a bigger step size for faster convergence. Since the EM algorithm is known as a bound optimizer[67] that can benefit from overrelaxation,[21] the EM-PCA can easily adopt overrelaxation. In addition to overrelaxation, the multigrid method is a general convergence acceleration method that quickly reduces persistent, hard-to-decay, low-frequency errors on a fine grid by turning them into easy-to-decay, high-frequency errors on a coarse grid. For instance, the multigrid was implemented for the EM algorithm in conjunction with a Poisson probability distribution for faster parameter estimation.[27] The synergetic effect of the combination of overrelaxation and the multigrid would be a good research topic, for it might show enhanced performance of the EM-PCA.

Last but not least, a valuable research topic would be the investigation of the potential benefits of the mixture of PCA models[81] for aerospace engineering applications. Tippling and Bishop devised the PCA mixture model as an amalgam of single PPCA models to yield a locally linear model for given observations. For an illustration, Figure 79 depicts the concept of the PCA mixture model compared to that of a single PCA model. With the help of the



(a) Single PCA model (b) PCA mixture model

Figure 79: Concept of single PCA and PCA mixture models

EM algorithm, the PCA mixture model can also automatically classify all observations into subgroups in which each PCA model describes its own separate region. Moreover, the EM algorithm naturally enables the PCA mixture model to estimate missing data through a mixture of PCA models. Since a combination of locally linear models is more sophisticated than a single globally monolithic, linear model, the PCA mixture model is expected to excel at addressing the applications of both POD and gappy POD. The following is a list of the abilities of the PCA mixture model associated with possible aerospace applications.

- Data classification
  - Domain decomposition for a reduced-order modeling of high-fidelity flow analysis[39,42]
  - Infrared camera image analysis for natural laminar flow monitoring[14]
- Local feature identification
  - Low-frequency vortex identification for experimental flow analysis
- Missing data estimation
  - Impaired data restoration for PIV and magnetic resonance imaging (MRI) measurements

APPENDIX A

PROOF

A.1 Matrix Identity

Theorem A.1.1 (Sherman–Morrison–Woodbury formula[15])**.** The matrix inversion lemma, Sherman–Morrison–Woodbury formula, or Woodbury formula is

$$(\mathbf{A}+\mathbf{UCV})^{-1} = \mathbf{A}^{-1} - \left[\mathbf{A}^{-1}\mathbf{U}\left(\mathbf{C}^{-1}+\mathbf{VA}^{-1}\mathbf{U}\right)^{-1}\mathbf{VA}^{-1}\right]$$

where $\mathbf{A}$ is an n by n, $\mathbf{U}$ is an n by k, $\mathbf{C}$ is a k by k, and $\mathbf{V}$ is a k by n matrix. In the special case where C is the identity matrix $\mathbf{I}$ and $\mathbf{V}$ is $\mathbf{V}^{\mathrm{T}}$, this identity reduces to

$$\left(\mathbf{A}+\mathbf{UV}^{\mathrm{T}}\right)^{-1} = \mathbf{A}^{-1} - \left[\mathbf{A}^{-1}\mathbf{U}\left(\mathbf{I}+\mathbf{V}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{U}\right)^{-1}\mathbf{V}^{\mathrm{T}}\mathbf{A}^{-1}\right] \tag{42}$$

Proposition A.1.1. For a rectangular matrix $\mathbf{W} \in \mathbb{R}^{d\times q}$,

$$\mathbf{W}^{\mathrm{T}}\left(\sigma^2\mathbf{I}+\mathbf{WW}^{\mathrm{T}}\right)^{-1} = \left(\sigma^2\mathbf{I}+\mathbf{W}^{\mathrm{T}}\mathbf{W}\right)^{-1}\mathbf{W}^{\mathrm{T}} \tag{43}$$

Proof. Let $\mathbf{A} = \sigma^2\mathbf{I}$, $\mathbf{U} = \mathbf{W}$, and $\mathbf{V} = \mathbf{W}$, then the inverse of a parenthesized term in the LHS of Eq. (43) can be rewritten with the help of Eq. (42) in Theorem A.1.1.

$$\begin{aligned}
\left(\sigma^2\mathbf{I}+\mathbf{WW}^{\mathrm{T}}\right)^{-1} &= \sigma^{-2}\left[\mathbf{I}-\sigma^{-2}\mathbf{W}\left(\mathbf{I}+\sigma^{-2}\mathbf{W}^{\mathrm{T}}\mathbf{W}\right)^{-1}\mathbf{W}^{\mathrm{T}}\right] \\
&= \sigma^{-2}\left[\mathbf{I}-\sigma^{-2}\mathbf{W}\left(\sigma^{-2}(\sigma^2\mathbf{I})+\sigma^{-2}\mathbf{W}^{\mathrm{T}}\mathbf{W}\right)^{-1}\mathbf{W}^{\mathrm{T}}\right] \\
&= \sigma^{-2}\left[\mathbf{I}-\mathbf{W}\left(\sigma^2\mathbf{I}+\mathbf{W}^{\mathrm{T}}\mathbf{W}\right)^{-1}\mathbf{W}^{\mathrm{T}}\right]
\end{aligned}$$

Thus, the LHS of the Eq. (43) comes to

$$\begin{aligned}
\mathbf{W}^{\mathrm{T}}\left(\sigma^2\mathbf{I}+\mathbf{WW}^{\mathrm{T}}\right)^{-1} &= \sigma^{-2}\left[\mathbf{W}^{\mathrm{T}}-\mathbf{W}^{\mathrm{T}}\mathbf{W}\left(\sigma^2\mathbf{I}+\mathbf{W}^{\mathrm{T}}\mathbf{W}\right)^{-1}\mathbf{W}^{\mathrm{T}}\right] \\
&= \sigma^{-2}\left[\mathbf{I}-\mathbf{W}^{\mathrm{T}}\mathbf{W}\left(\sigma^2\mathbf{I}+\mathbf{W}^{\mathrm{T}}\mathbf{W}\right)^{-1}\right]\mathbf{W}^{\mathrm{T}}
\end{aligned} \tag{44}$$

From the equality of the Eq. (43) and the Eq. (44), it is necessary to show the following equality.

$$\sigma^{-2}\left[\mathbf{I}-\mathbf{W}^{\mathrm{T}}\mathbf{W}\left(\sigma^2\mathbf{I}+\mathbf{W}^{\mathrm{T}}\mathbf{W}\right)^{-1}\right]\mathbf{W}^{\mathrm{T}} = \left(\sigma^2\mathbf{I}+\mathbf{W}^{\mathrm{T}}\mathbf{W}\right)^{-1}\mathbf{W}^{\mathrm{T}} \tag{45}$$

For a term in the bracket of the LHS of the Eq. (45), let $\mathbf{A} = \mathbf{W}^{\mathrm{T}}\mathbf{W}$ and $\mathbf{B} = \left(\sigma^2\mathbf{I} + \mathbf{W}^{\mathrm{T}}\mathbf{W}\right)$ and use a matrix inversion identity $\mathbf{AB}^{-1} = (\mathbf{BA}^{-1})^{-1}$.

$$\mathbf{W}^{\mathrm{T}}\mathbf{W}\left(\sigma^2\mathbf{I} + \mathbf{W}^{\mathrm{T}}\mathbf{W}\right)^{-1} = \left[\left(\sigma^2\mathbf{I} + \mathbf{W}^{\mathrm{T}}\mathbf{W}\right)\left(\mathbf{W}^{\mathrm{T}}\mathbf{W}\right)^{-1}\right]^{-1} = \left[\mathbf{I} + \sigma^2\left(\mathbf{W}^{\mathrm{T}}\mathbf{W}\right)^{-1}\right]^{-1}$$

Thus, the LHS of the Eq. (45) turns into

$$\sigma^{-2}\left[\mathbf{I} - \mathbf{W}^{\mathrm{T}}\mathbf{W}\left(\sigma^2\mathbf{I} + \mathbf{W}^{\mathrm{T}}\mathbf{W}\right)^{-1}\right]\mathbf{W}^{\mathrm{T}} = \sigma^{-2}\left\{\mathbf{I} - \left[\mathbf{I} + \sigma^2\left(\mathbf{W}^{\mathrm{T}}\mathbf{W}\right)^{-1}\right]^{-1}\right\}\mathbf{W}^{\mathrm{T}} \quad (46)$$

In order to rephrase the braced term in the RHS of the Eq. (46), let $\mathbf{A} = \mathbf{I} + \sigma^2\left(\mathbf{W}^{\mathrm{T}}\mathbf{W}\right)^{-1}$, and use a matrix inversion identity $\mathbf{I} - \mathbf{A}^{-1} = (\mathbf{A} - \mathbf{I})\mathbf{A}^{-1}$.

$$\begin{aligned}
\mathbf{I} - \left[\mathbf{I} + \sigma^2\left(\mathbf{W}^{\mathrm{T}}\mathbf{W}\right)^{-1}\right]^{-1} &= \left[\mathbf{I} + \sigma^2\left(\mathbf{W}^{\mathrm{T}}\mathbf{W}\right)^{-1} - \mathbf{I}\right]\left[\mathbf{I} + \sigma^2\left(\mathbf{W}^{\mathrm{T}}\mathbf{W}\right)^{-1}\right]^{-1} \\
&= \sigma^2\left(\mathbf{W}^{\mathrm{T}}\mathbf{W}\right)^{-1}\left[\mathbf{I} + \sigma^2\left(\mathbf{W}^{\mathrm{T}}\mathbf{W}\right)^{-1}\right]^{-1} \\
&= \sigma^2\left(\mathbf{W}^{\mathrm{T}}\mathbf{W}\right)^{-1}\left(\mathbf{W}^{\mathrm{T}}\mathbf{W}\right)\left(\sigma^2\mathbf{I} + \mathbf{W}^{\mathrm{T}}\mathbf{W}\right)^{-1} \\
&= \sigma^2\left(\sigma^2\mathbf{I} + \mathbf{W}^{\mathrm{T}}\mathbf{W}\right)^{-1}
\end{aligned}$$

In the above derivation, the bracketed term in the RHS of the Eq. (46) is manipulated by letting $\mathbf{A} = \sigma^2\mathbf{I}$ and $\mathbf{B} = \mathbf{W}^{\mathrm{T}}\mathbf{W}$ and using a matrix inversion identity $(\mathbf{A} + \mathbf{B})^{-1} = \mathbf{B}(\mathbf{A} + \mathbf{B})^{-1}\mathbf{A}$.

$$\begin{aligned}
\left[\mathbf{I} + \sigma^2\left(\mathbf{W}^{\mathrm{T}}\mathbf{W}\right)^{-1}\right]^{-1} = \sigma^{-2}\left[\left(\sigma^2\mathbf{I}\right)^{-1} + \left(\mathbf{W}^{\mathrm{T}}\mathbf{W}\right)^{-1}\right]^{-1} &= \sigma^{-2}\left(\mathbf{W}^{\mathrm{T}}\mathbf{W}\right)\left(\sigma^2\mathbf{I} + \mathbf{W}^{\mathrm{T}}\mathbf{W}\right)^{-1}\left(\sigma^2\mathbf{I}\right) \\
&= \left(\mathbf{W}^{\mathrm{T}}\mathbf{W}\right)\left(\sigma^2\mathbf{I} + \mathbf{W}^{\mathrm{T}}\mathbf{W}\right)^{-1}
\end{aligned}$$

Therefore, Eq. (46) is reduced to the following form, which proves the Eq. (45) and concludes the proof.

$$\sigma^{-2}\left[\mathbf{I} - \mathbf{W}^{\mathrm{T}}\mathbf{W}\left(\sigma^2\mathbf{I} + \mathbf{W}^{\mathrm{T}}\mathbf{W}\right)^{-1}\right]\mathbf{W}^{\mathrm{T}} = \left(\sigma^2\mathbf{I} + \mathbf{W}^{\mathrm{T}}\mathbf{W}\right)^{-1}\mathbf{W}^{\mathrm{T}}$$

□

Corollary A.1.1.

$$\mathbf{W}\left(\sigma^2\mathbf{I} + \mathbf{W}^{\mathrm{T}}\mathbf{W}\right)^{-1} = \left(\sigma^2\mathbf{I} + \mathbf{W}\mathbf{W}^{\mathrm{T}}\right)^{-1}\mathbf{W}$$

Proof. Let $\mathbf{W}^{\mathrm{T}} = \mathbf{W}$ in Proposition A.1.1. □

APPENDIX B

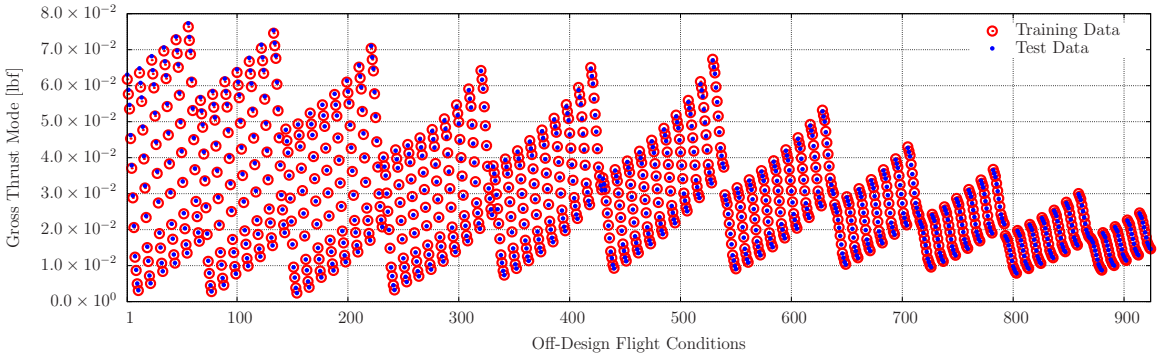
SUPPLEMENTS FOR REDUCED-ORDER NPSS MODELING

B.1 Validation of the Bases and Coefficients of Engine Deck Responses

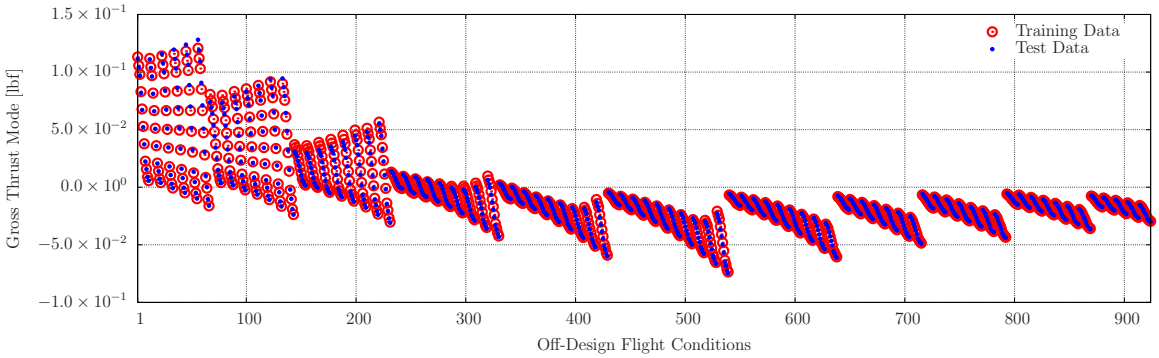
For all engine deck responses discussed in Chapter 6, the qualities of their modes and coefficients are delineated in Figures 80–83 and Figures 84–87, respectively. Note that the first modes and corresponding the first coefficients are of paramount importance since the first modes account for over 90% of total variations in engine deck responses as implied by their eigenvalues in Table 9. As depicted in Figures 80(a)–83(a), the first modes obtained from the training data are virtually identical to those achieved from the test data. Similarly, Figures 84(a)–87(a) show the exceptional fitness of the first coefficients for the test data as indicated by their R^2 values.

B.2 Worst Prediction Results of Engine Deck Responses

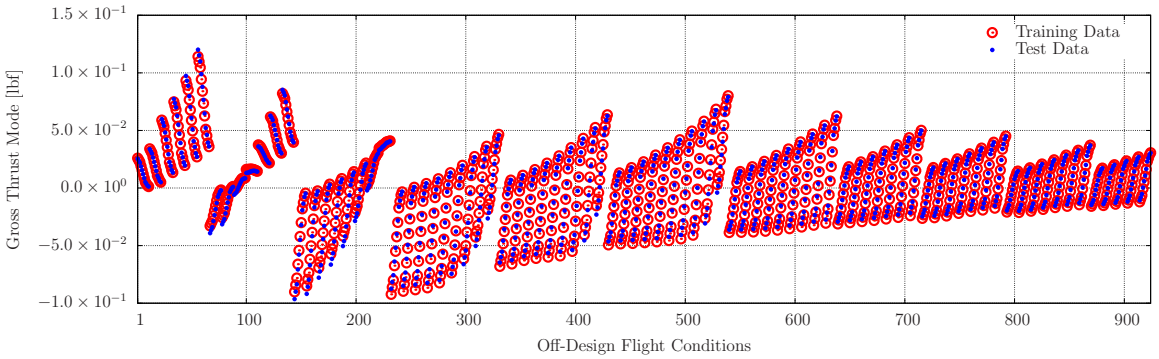
As an illustration of predicted engine deck responses, the worst prediction results are presented in Figures 88 and 89 in terms of an R^2 and a maximum NRSE, respectively. Since even the lowest R^2 values are around 0.999, estimated engine deck responses align well with the true values in Figure 88, and likewise, despite the highest NRSEs being around 35.16% at most, predicted engine deck responses containing maximum NRSEs show exceptional agreement with the exact values in Figure 89 because of their considerably high R^2 values. For a detailed view of the regions where the highest NRSE occurs, Figure 90 provides zoomed-in plots of Figure 89. Moreover, Figures 88 and 89 are re-plotted with respect to changes in the Mach numbers at the minimum and maximum throttle settings in Figures 91 and 92, respectively.



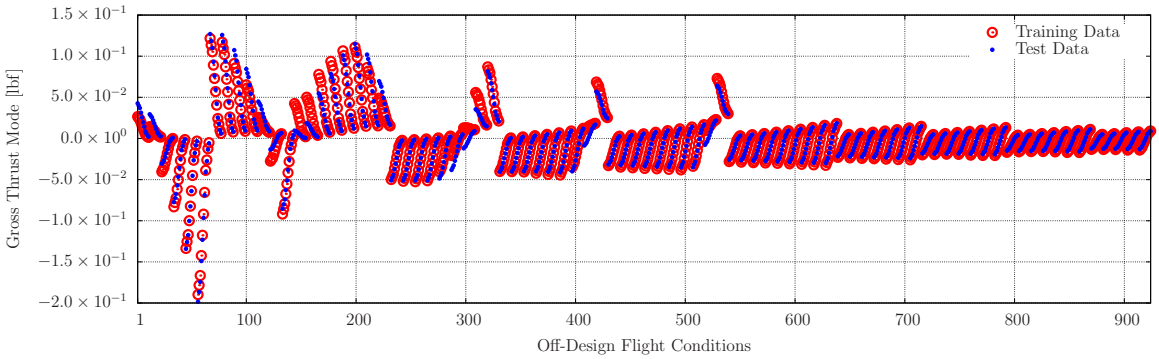
(a) 1st mode



(b) 2nd mode

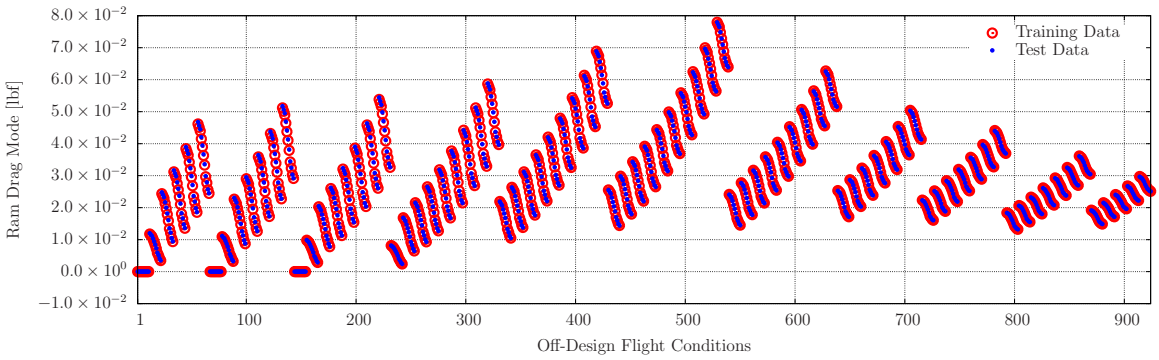


(c) 3rd mode

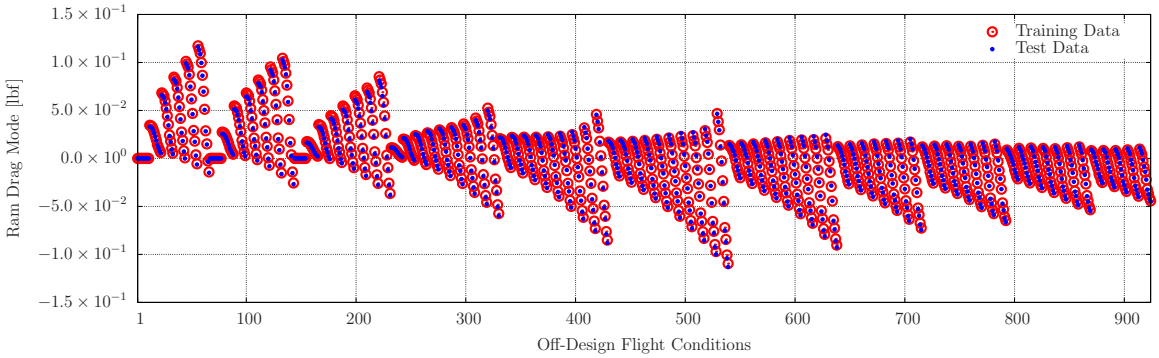


(d) 4th mode

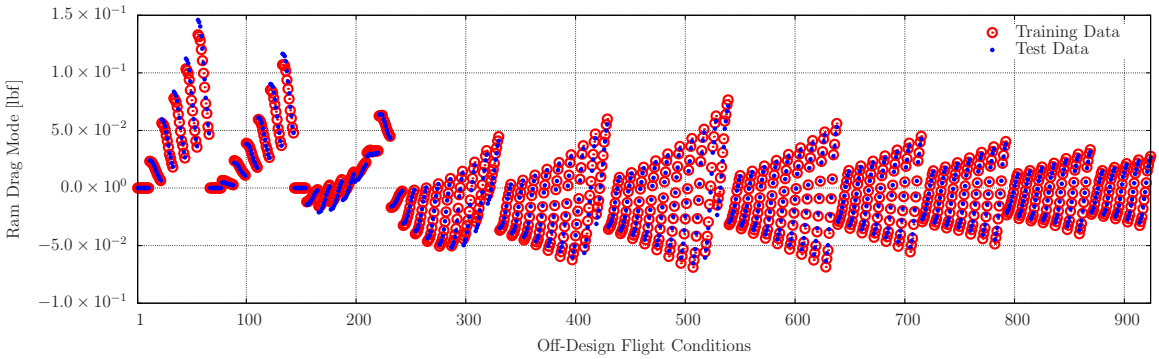
Figure 80: Modes of gross thrust obtained with training and test data



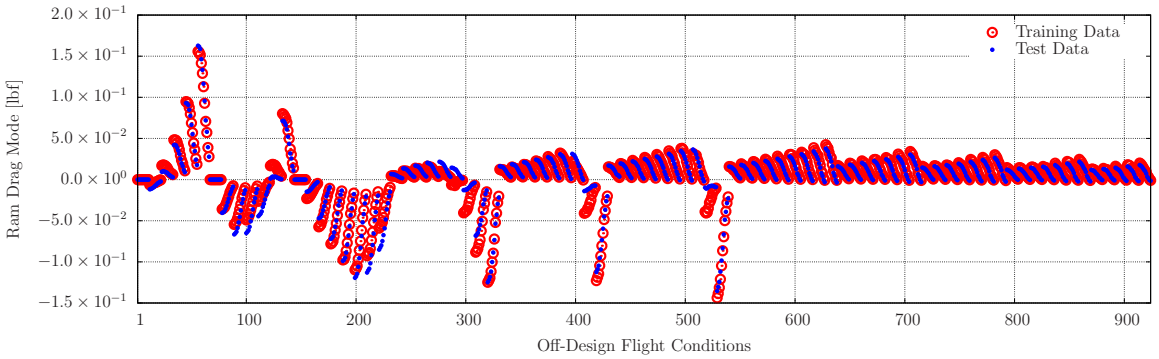
(a) 1st mode



(b) 2nd mode

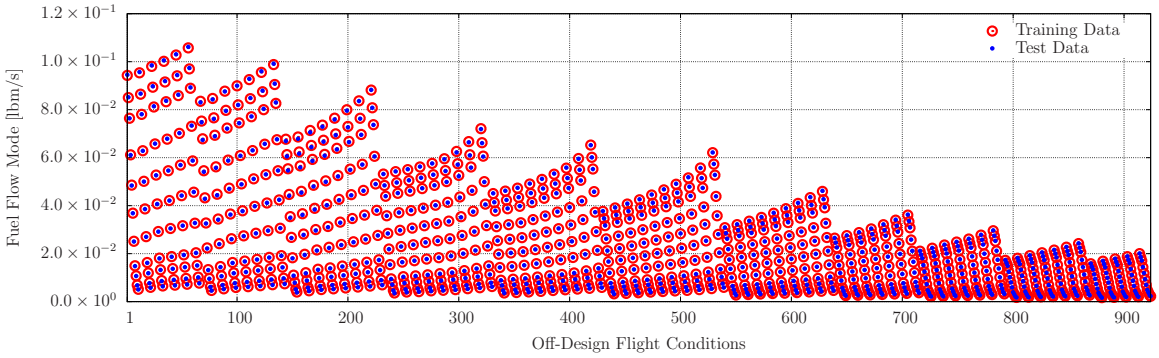


(c) 3rd mode

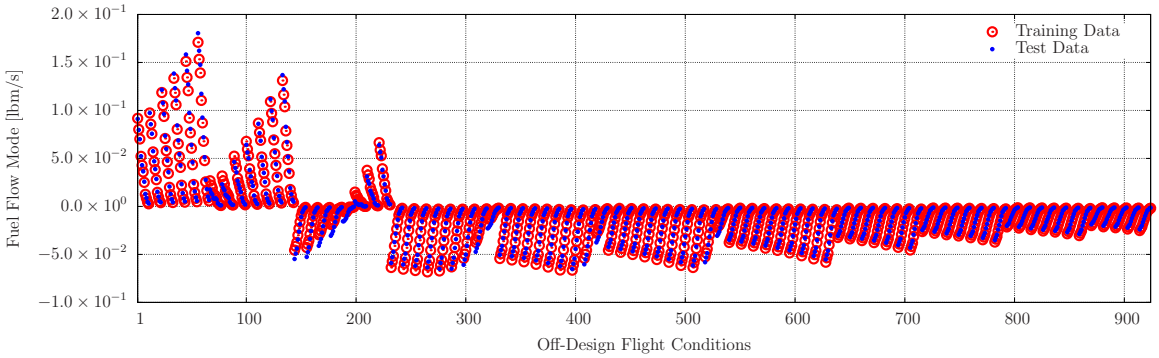


(d) 4th mode

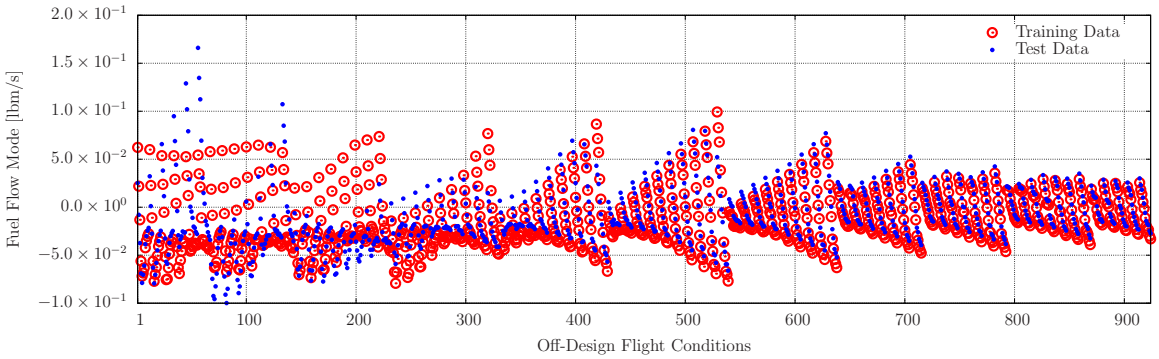
Figure 81: Modes of ram drag obtained with training and test data



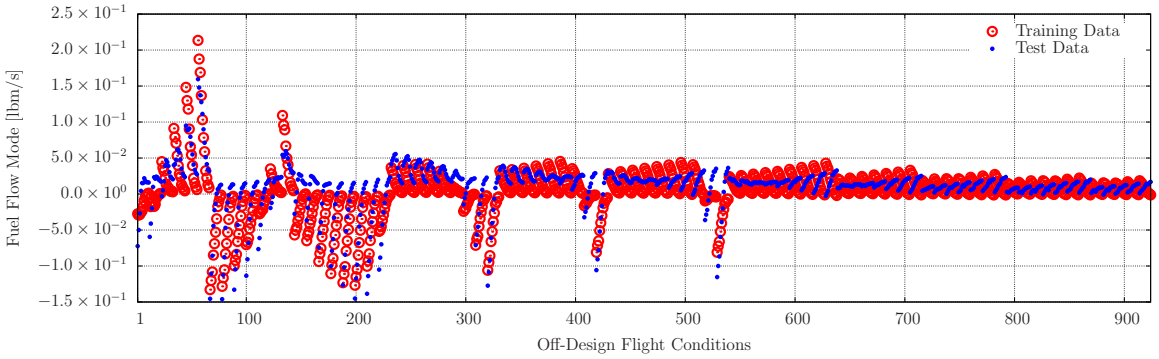
(a) 1st mode



(b) 2nd mode

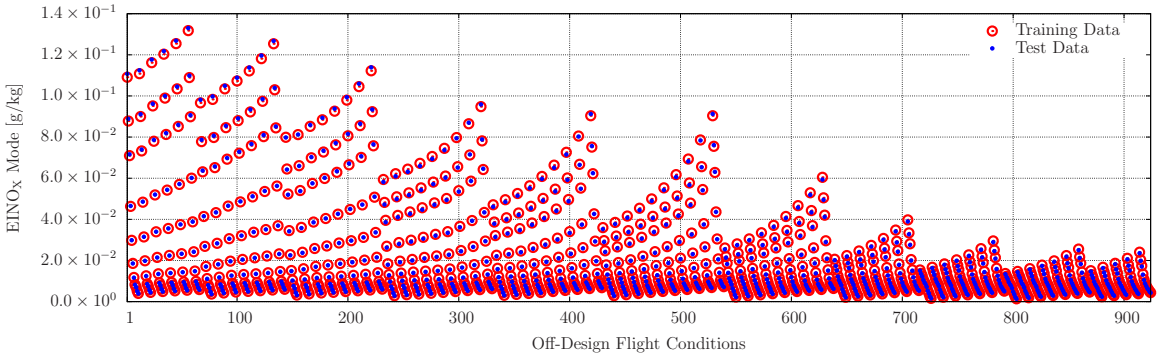


(c) 3rd mode

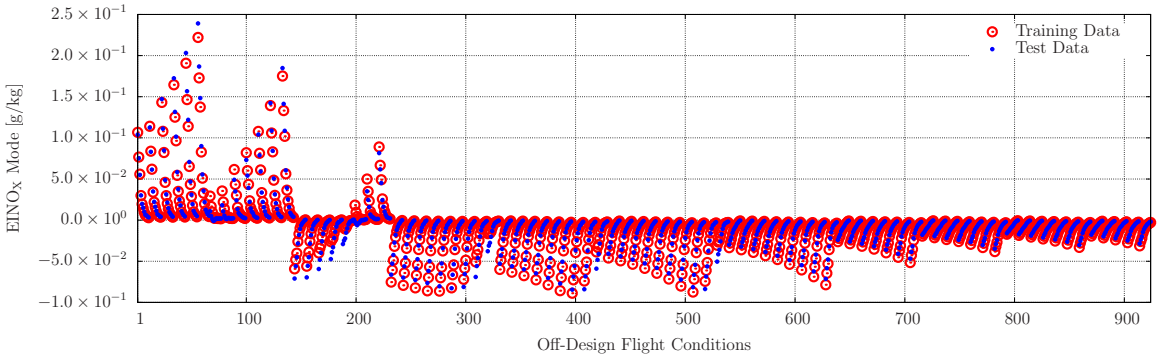


(d) 4th mode

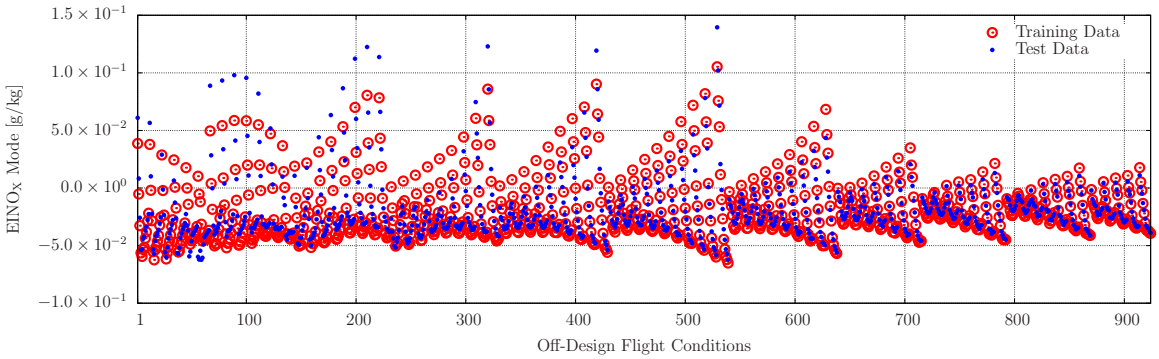
Figure 82: Modes of fuel flow obtained with training and test data



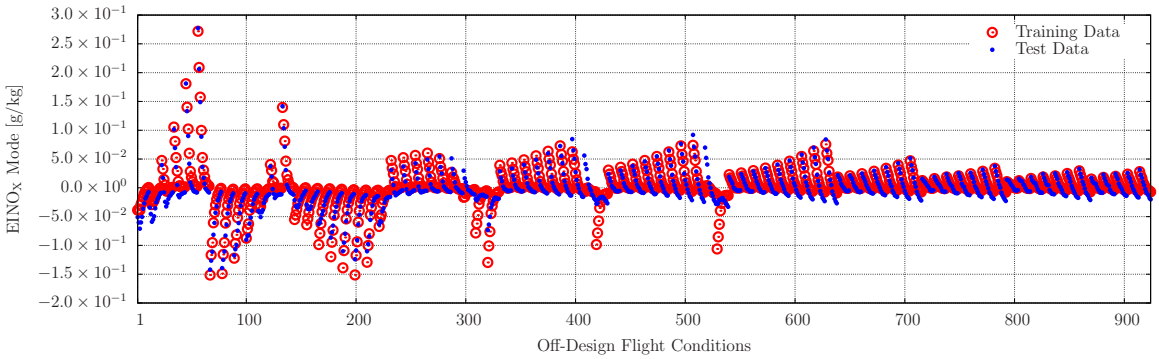
(a) 1st mode



(b) 2nd mode

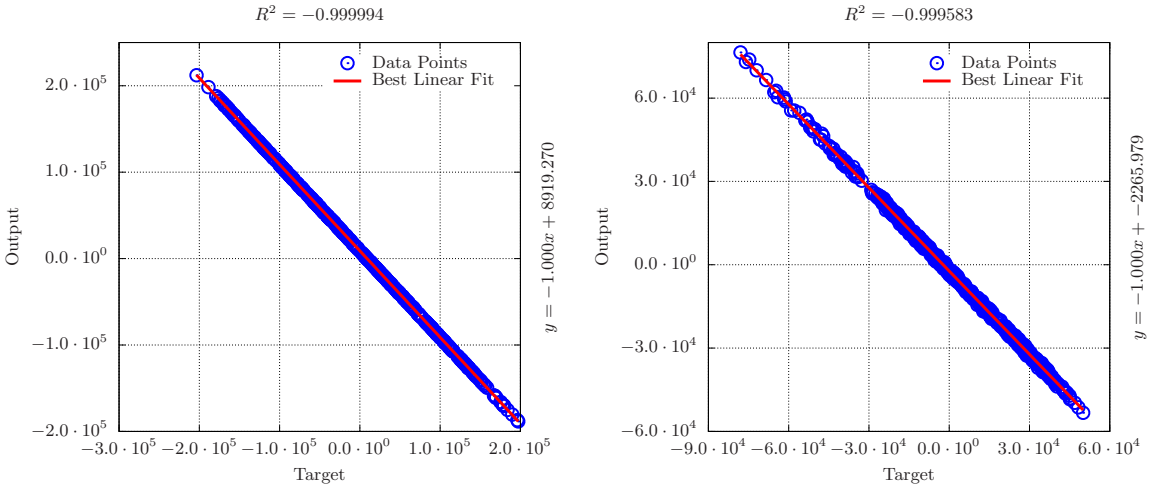


(c) 3rd mode



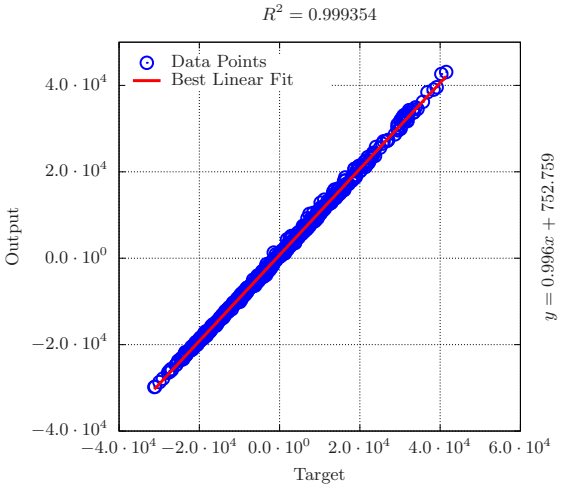
(d) 4th mode

Figure 83: Modes of $EINO_X$ obtained with training and test data

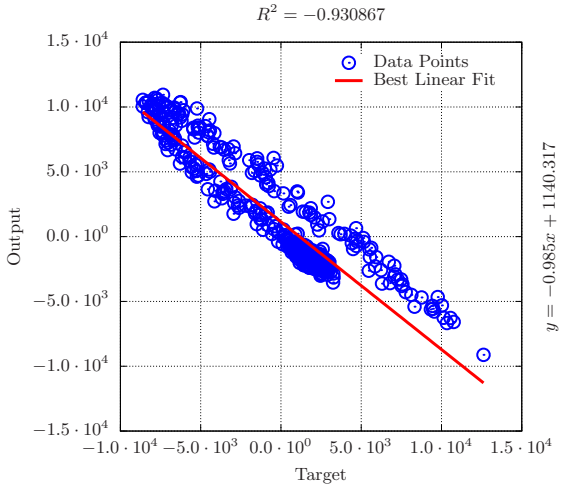


(a) 1st coefficient

(b) 2nd coefficient

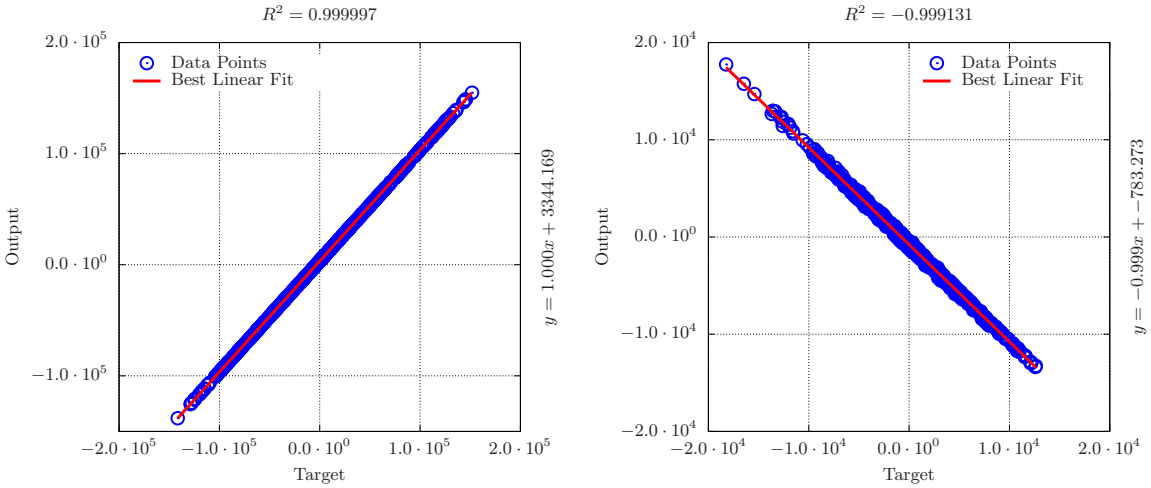


(c) 3rd coefficient



(d) 4th coefficient

Figure 84: R^2 plots of the weighting coefficients of gross thrust



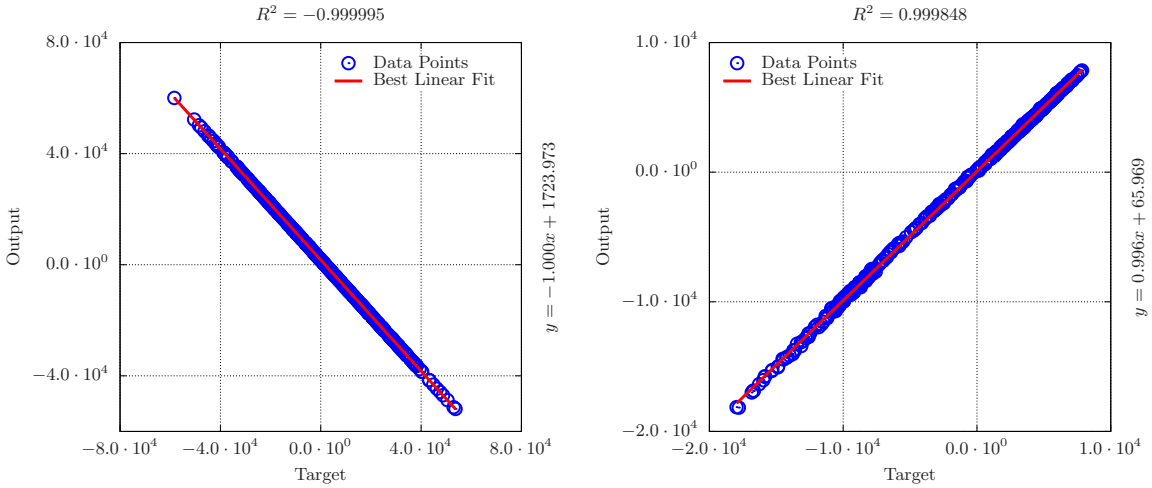
(a) 1st coefficient

(b) 2nd coefficient

(c) 3rd coefficient

(d) 4th coefficient

Figure 85: R^2 plots of the weighting coefficients of ram drag



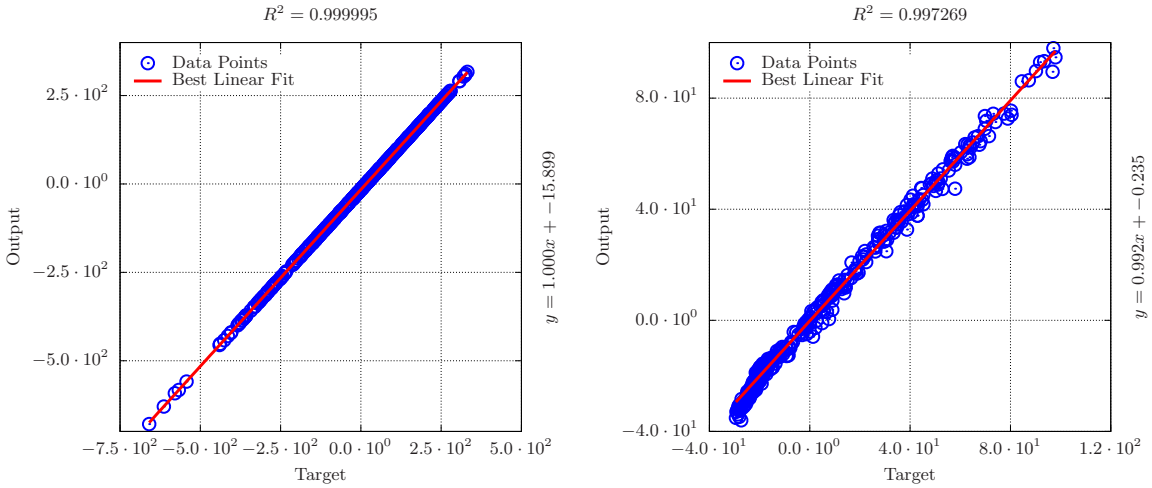
(a) 1st coefficient

(b) 2nd coefficient

(c) 3rd coefficient

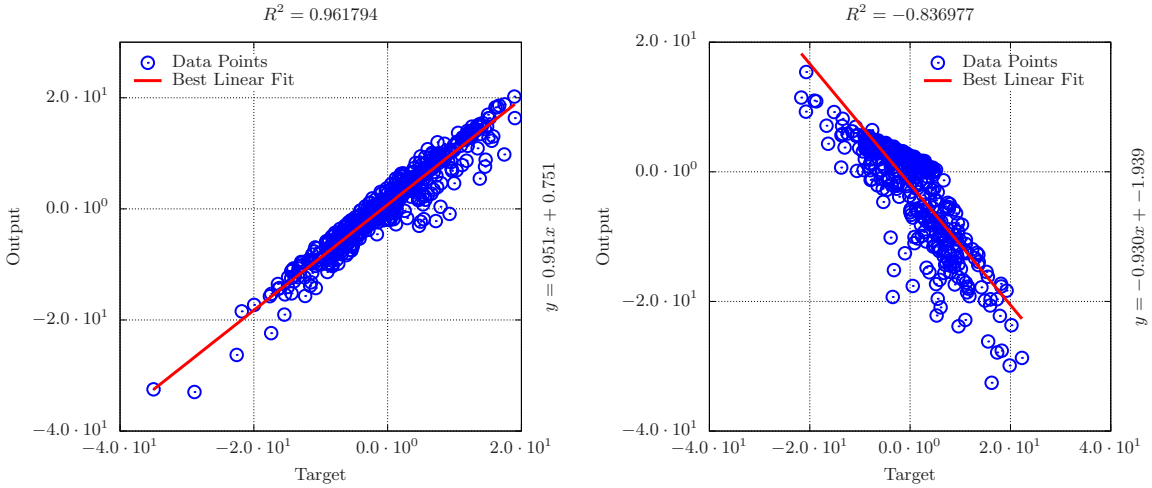
(d) 4th coefficient

Figure 86: R^2 plots of the weighting coefficients of fuel flow



(a) 1st coefficient

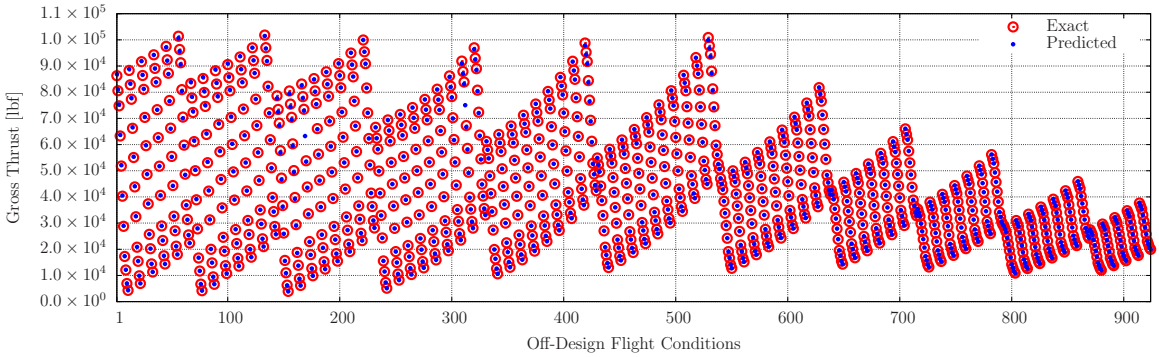
(b) 2nd coefficient



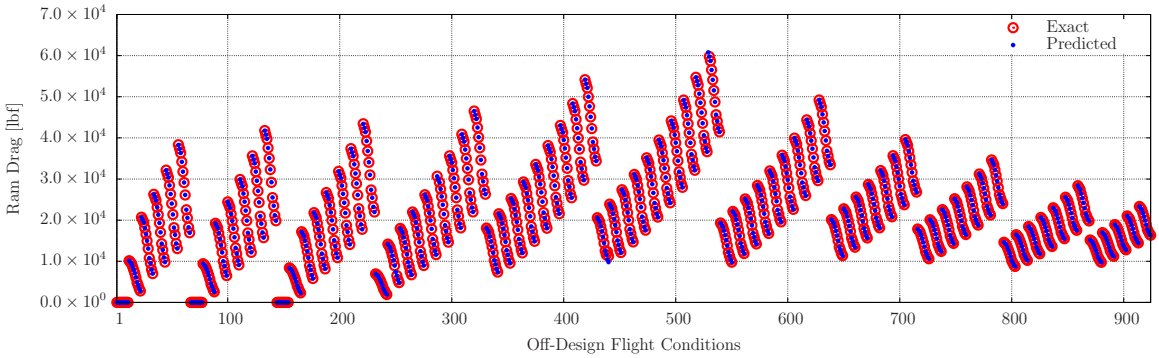
(c) 3rd coefficient

(d) 4th coefficient

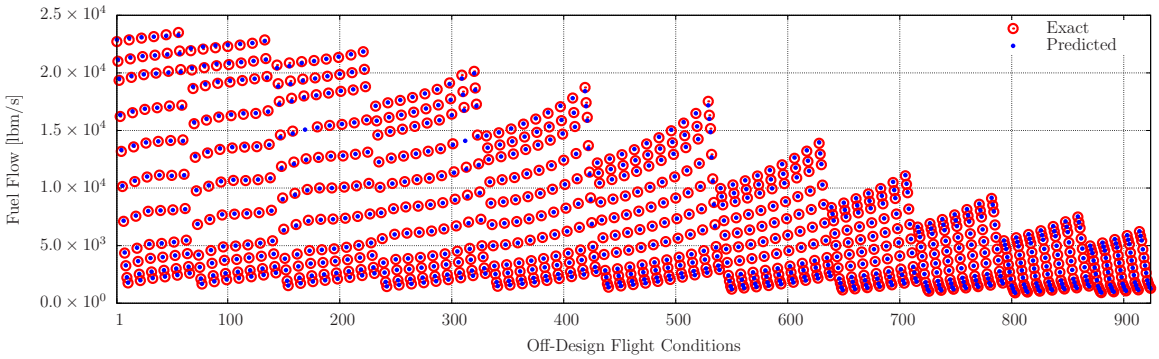
Figure 87: R^2 plots of the weighting coefficients of EINO_X



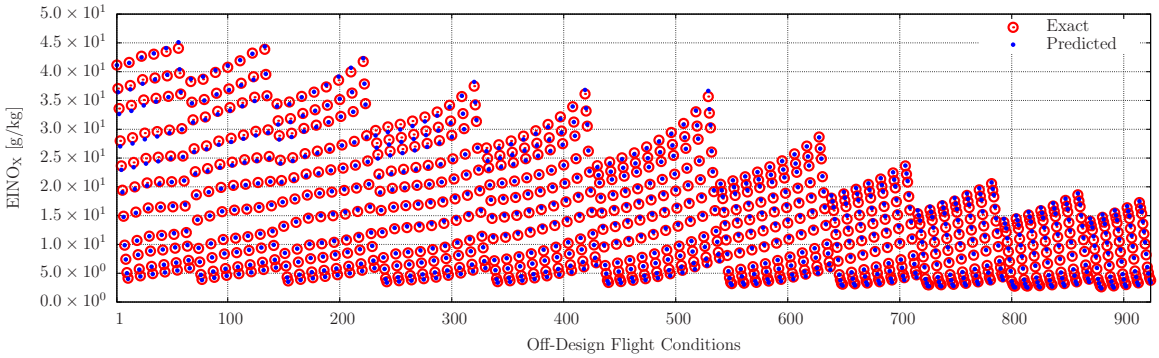
(a) Gross thrust of the 324^{th} test engine deck: $R^2 = 0.9999291$



(b) Ram drag of the 289^{th} test engine deck: $R^2 = 0.9999515$

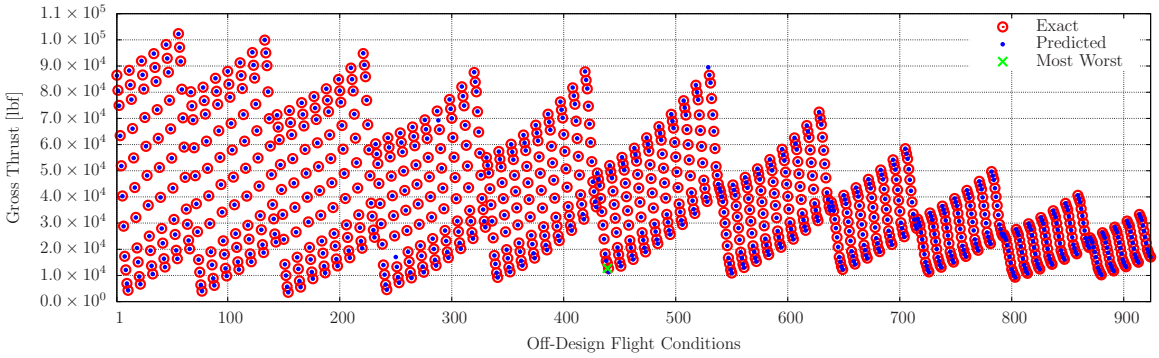


(c) Fuel flow of the 324^{th} test engine deck: $R^2 = 0.9998904$

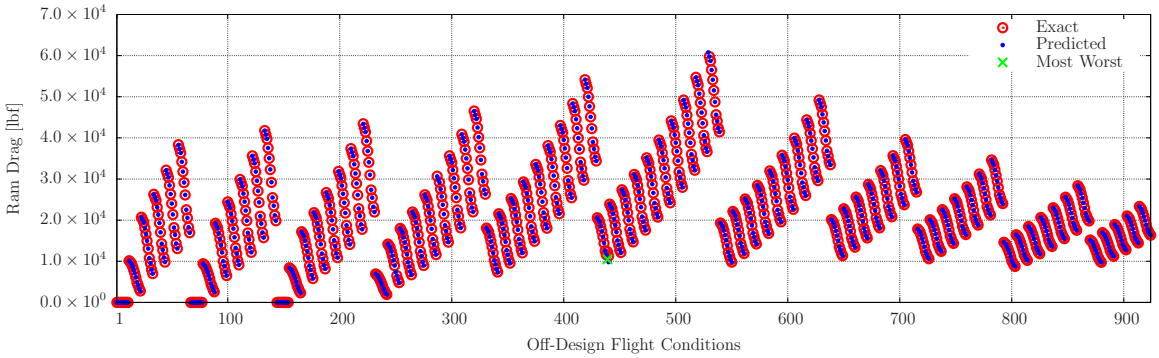


(d) EINO_X of the 117^{th} test engine deck: $R^2 = 0.9987783$

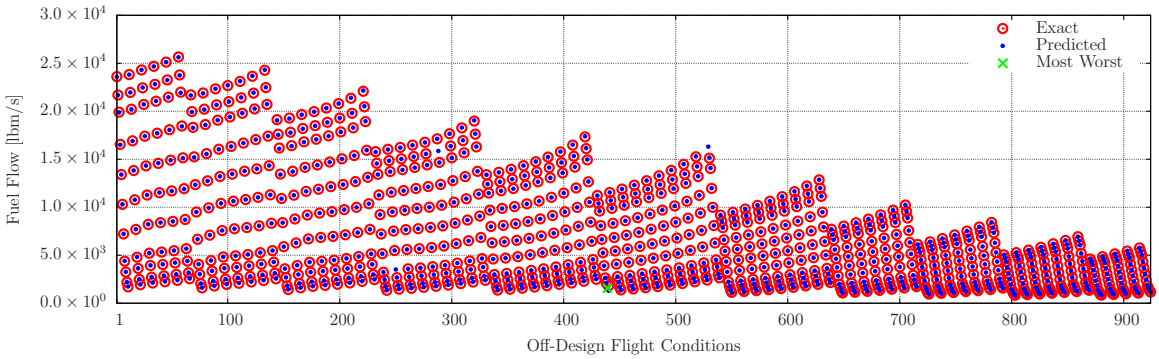
Figure 88: Actual and predicted engine deck responses: the worst R^2



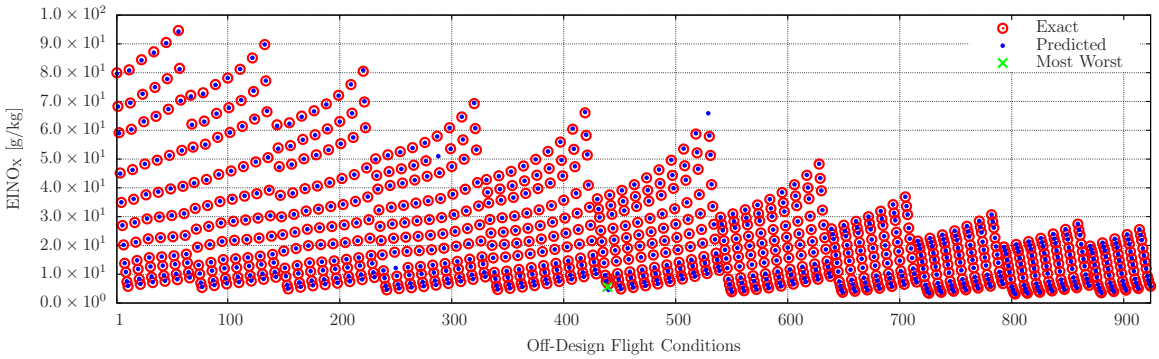
(a) Gross thrust of the 289$^{\text{th}}$ test engine deck: maximum NRSE = 12.91273%, $R^2 = 0.999982$



(b) Ram drag of the 289$^{\text{th}}$ test engine deck: maximum NRSE = 15.18704%, $R^2 = 0.9999515$

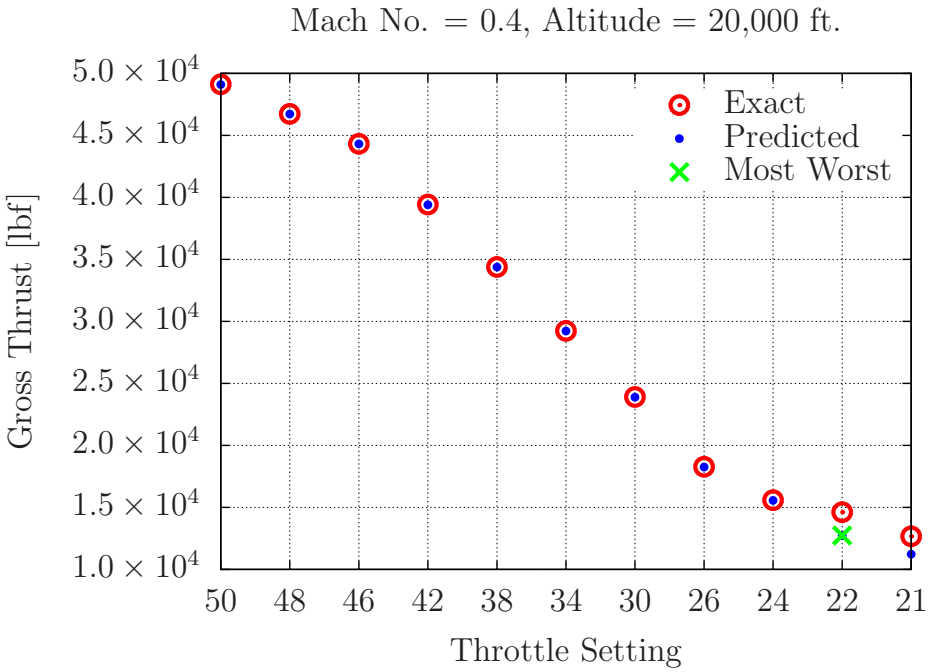


(c) Fuel flow of the 289$^{\text{th}}$ test engine deck: maximum NRSE = 31.61031%, $R^2 = 0.9999653$

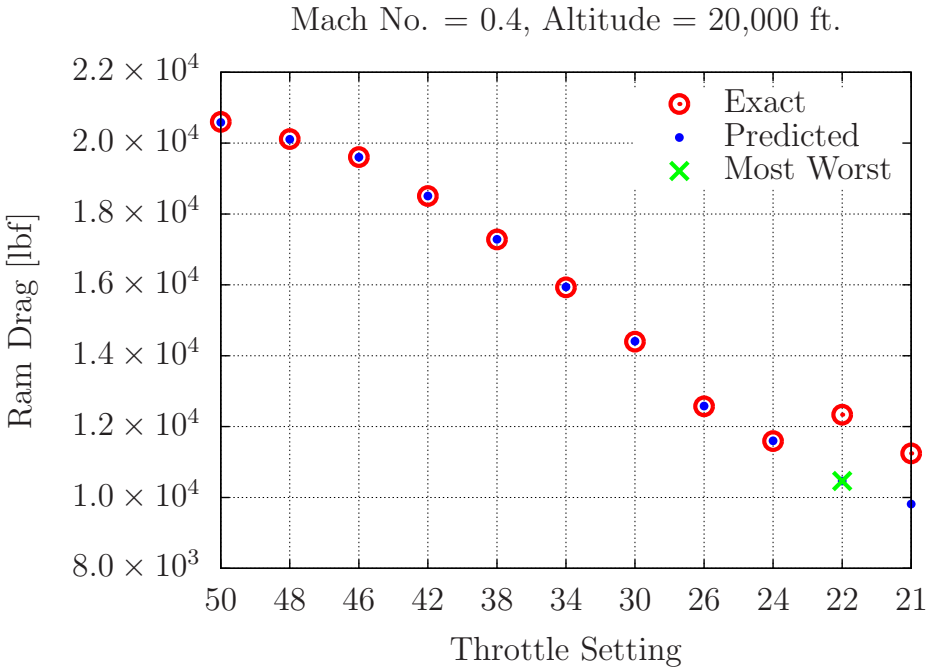


(d) EINO_X of the 289$^{\text{th}}$ test engine deck: maximum NRSE = 35.16145%, $R^2 = 0.9999305$

Figure 89: Actual and predicted engine deck responses: the maximum NRSE

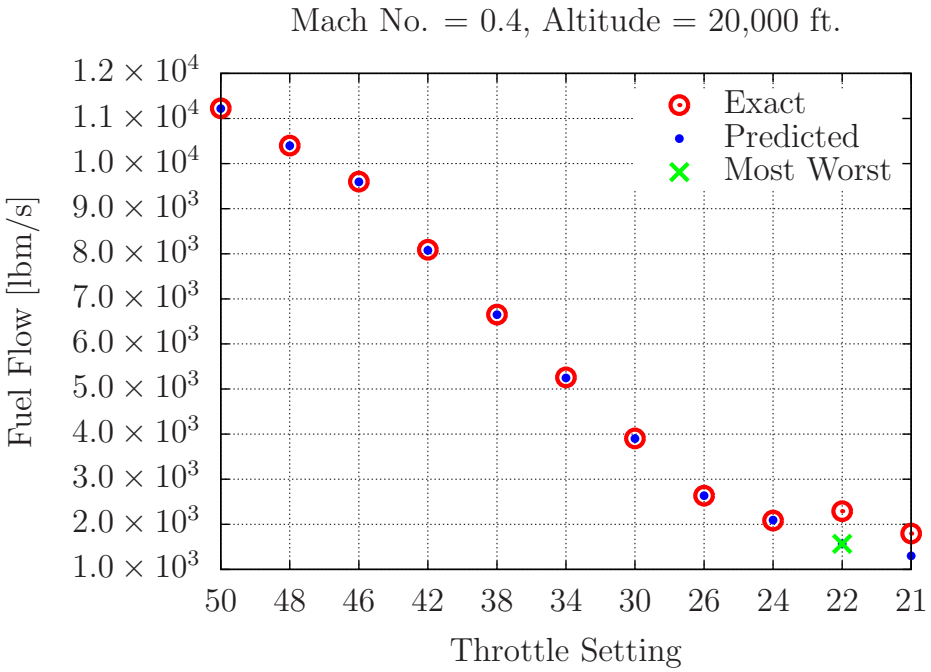


(a) Gross thrust: maximum NRSE = 12.91273%

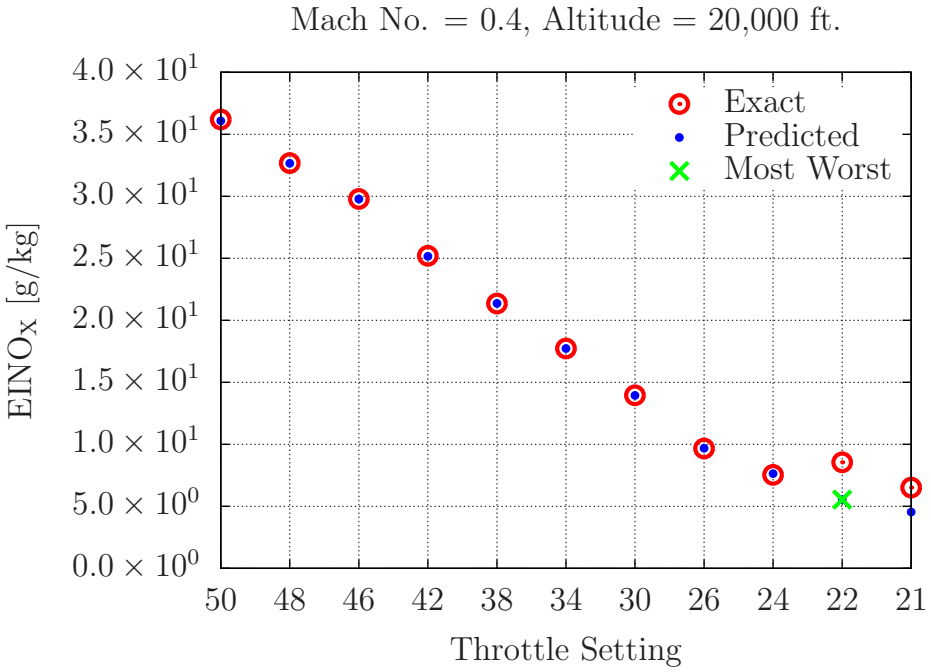


(b) Ram drag: maximum NRSE = 15.18704%

Figure 90: Zoomed-in actual and predicted engine deck responses: the maximum NRSE

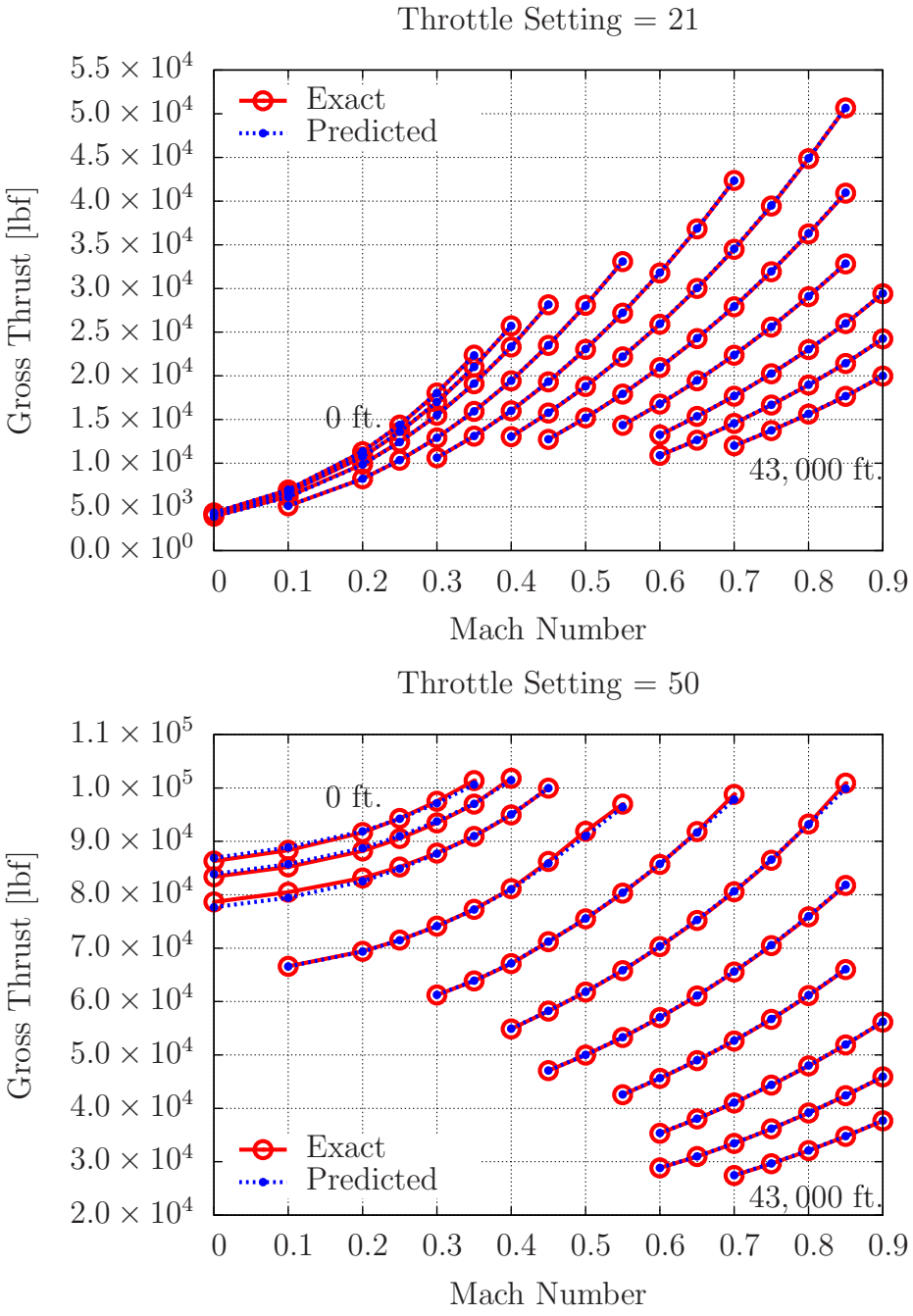


(c) Fuel flow: maximum NRSE = 31.61031%



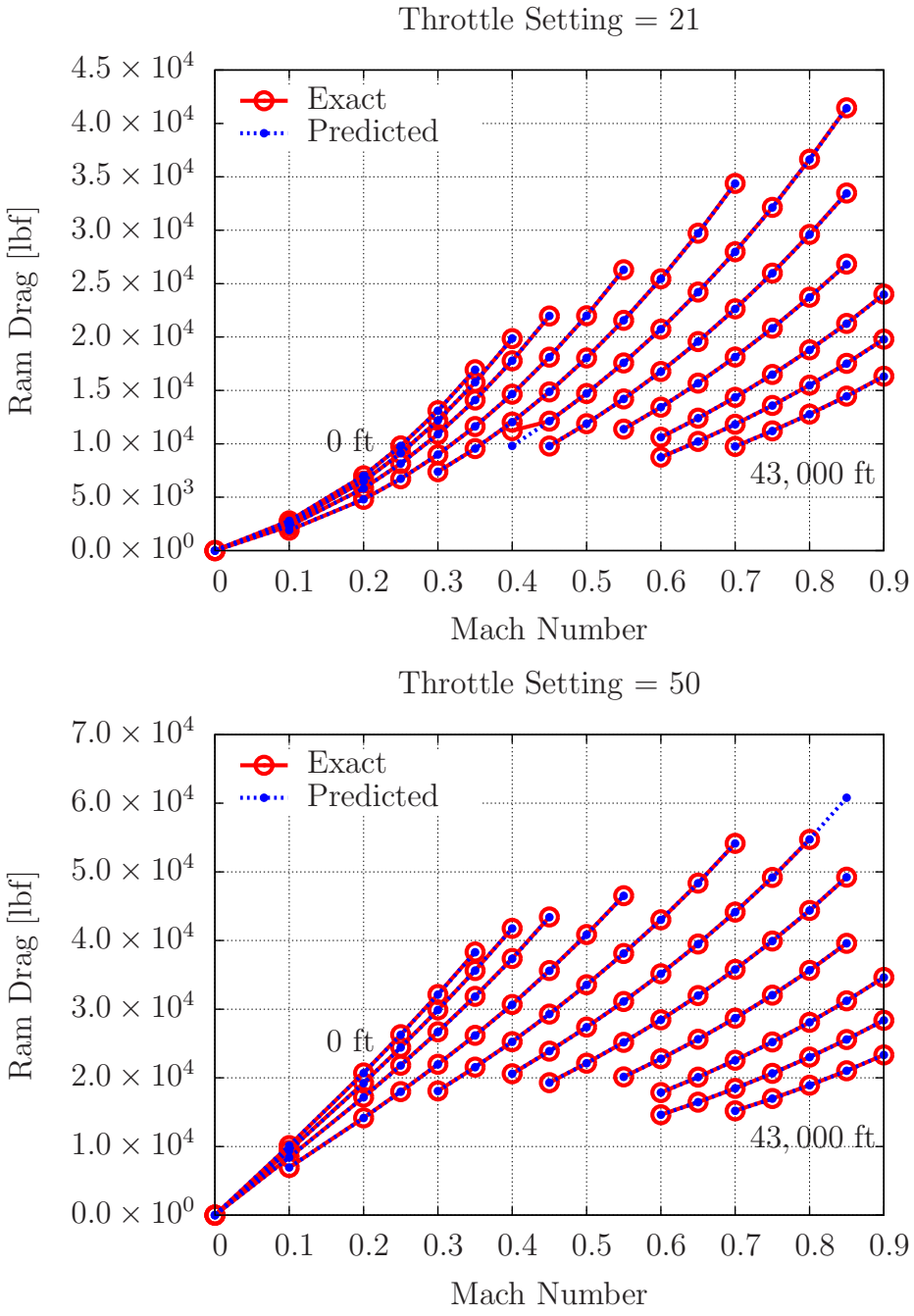
(d) $EINO_X$: maximum NRSE = 35.16145%

Figure 90: Zoomed-in actual and predicted engine deck responses: the maximum NRSE



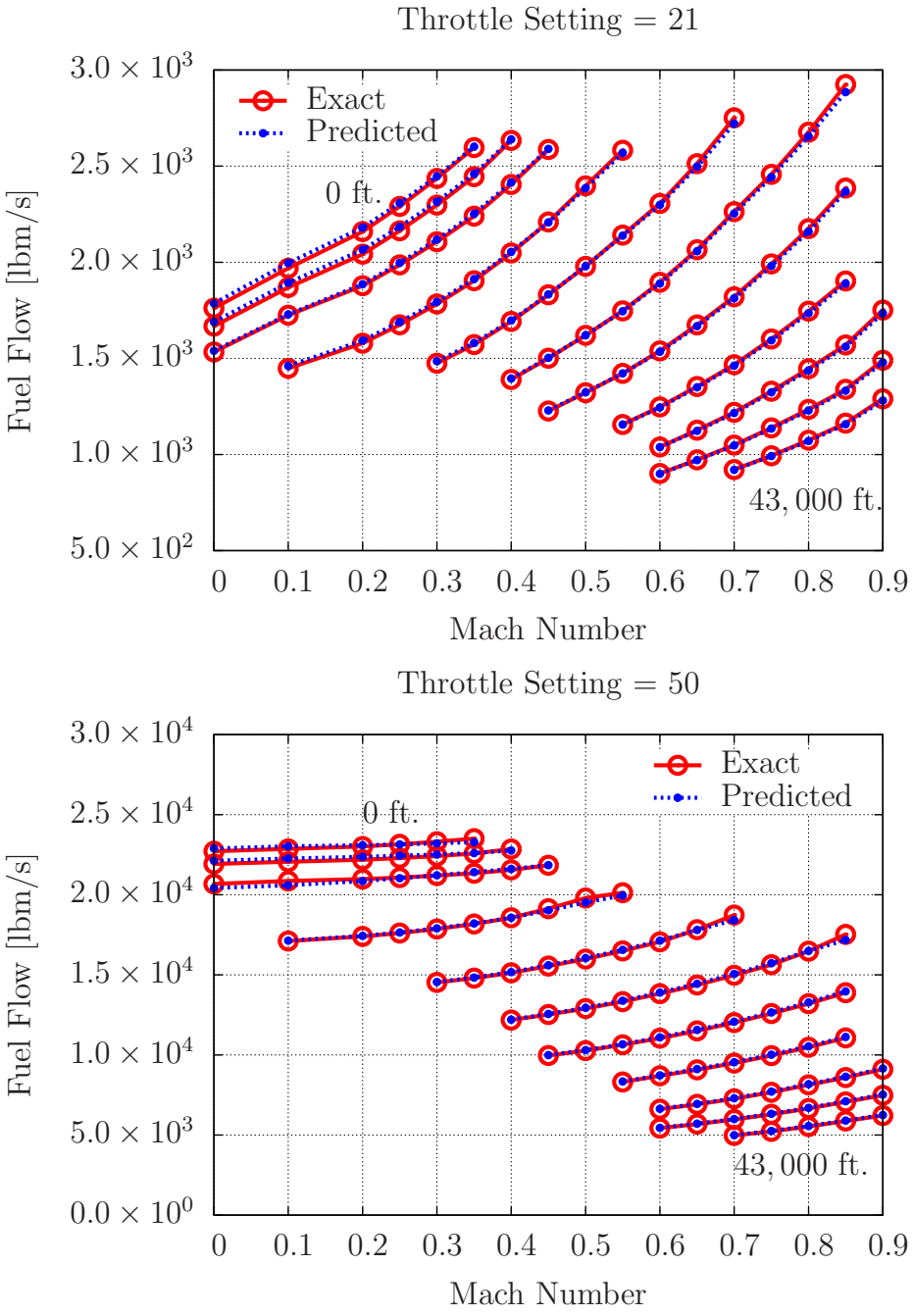
(b) Gross thrust of the 324$^{\text{th}}$ test engine deck: $R^2 = 0.9999291$

Figure 91: Actual and predicted engine deck responses with a Mach number: the worst R^2



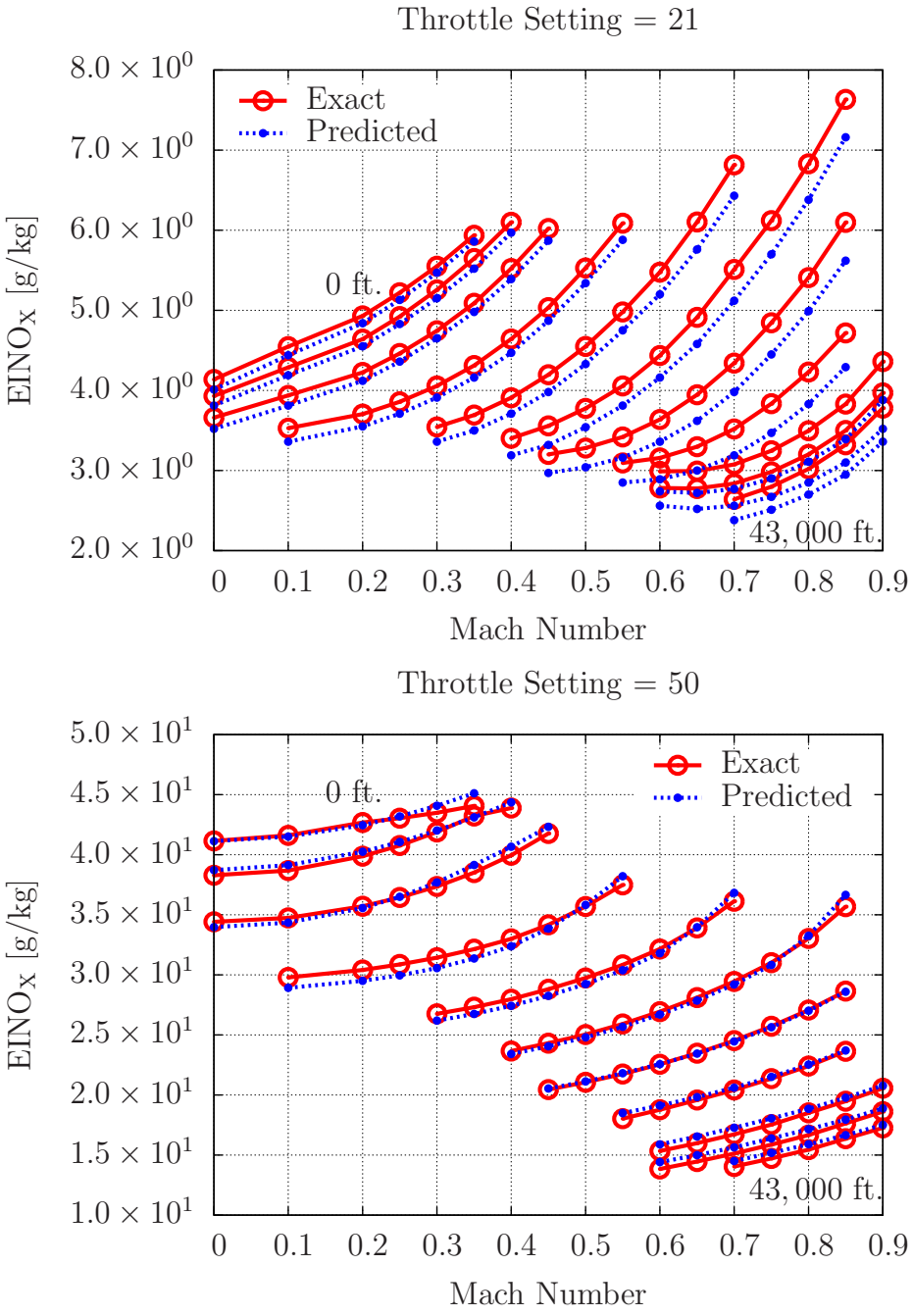
(d) Ram drag of the 289$^{\text{th}}$ test engine deck: $R^2 = 0.9999515$

Figure 91: Actual and predicted engine deck responses with a Mach number: the worst R^2



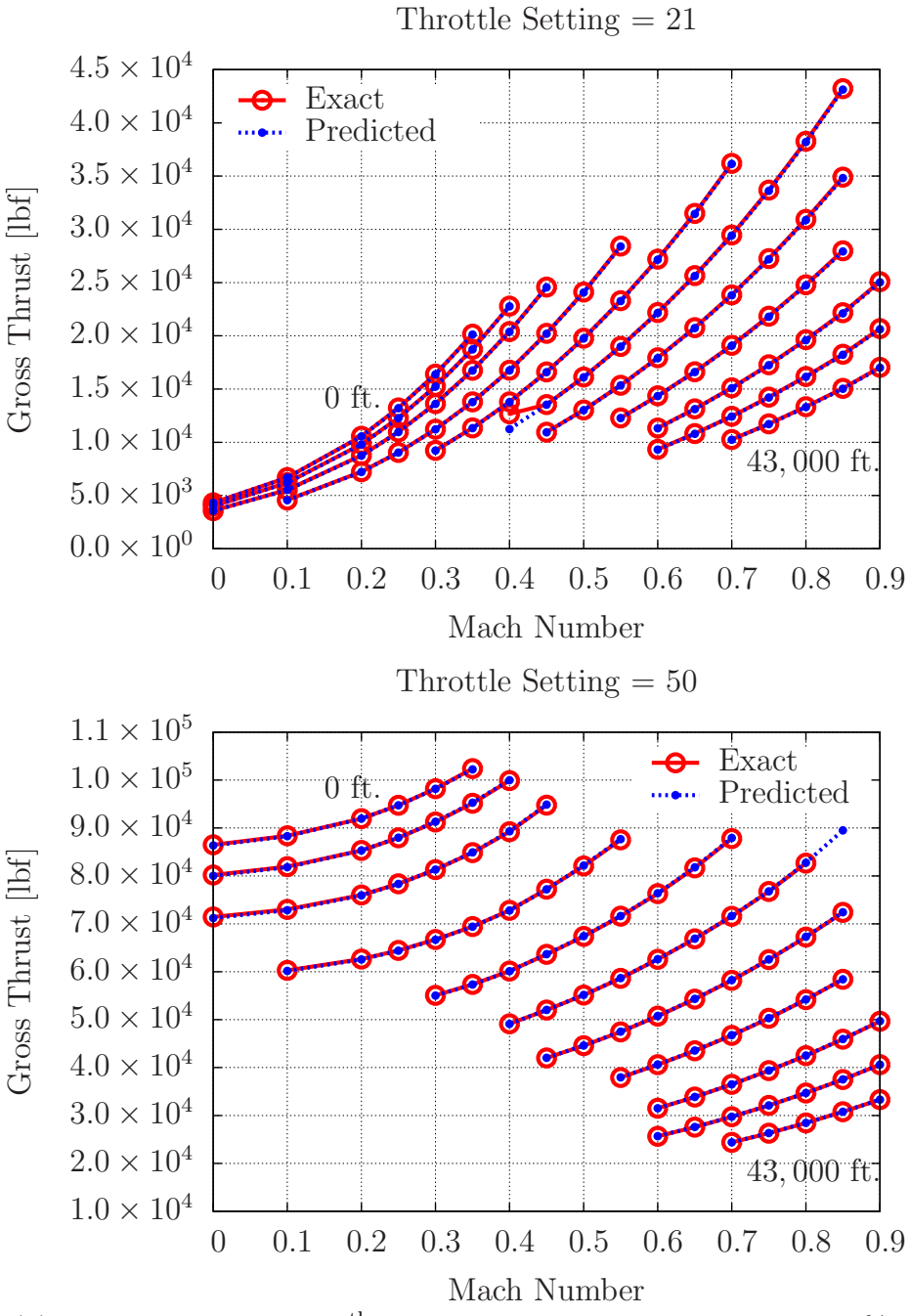
(f) Fuel flow of the 324^{th} test engine deck: $R^2 = 0.9998904$

Figure 91: Actual and predicted engine deck responses with a Mach number: the worst R^2



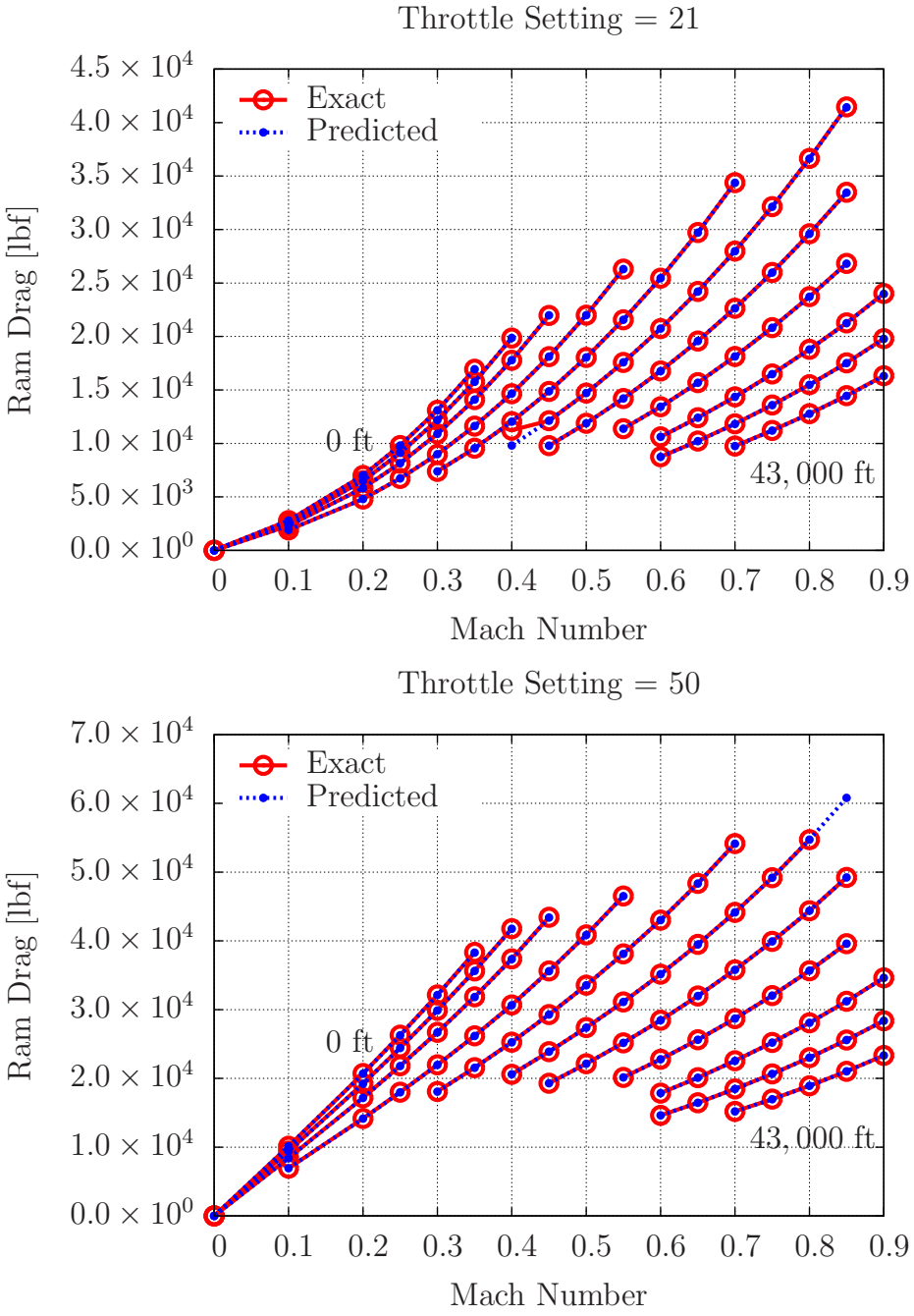
(h) $\mathrm{EINO_X}$ of the 117^{th} test engine deck: $R^2 = 0.9987783$

Figure 91: Actual and predicted engine deck responses with a Mach number: the worst R^2



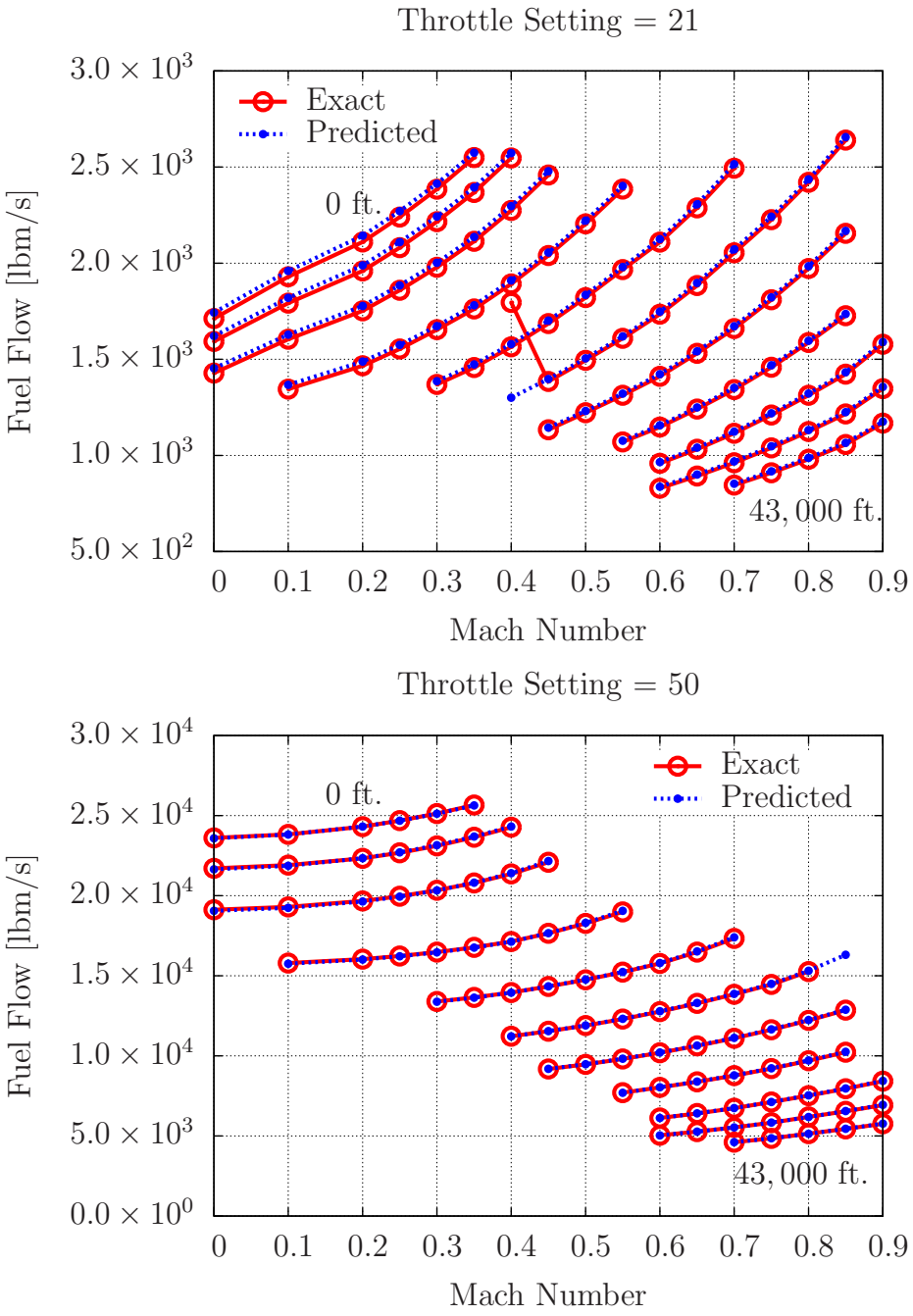
(b) Gross thrust of the 289$^{\text{th}}$ test engine deck: NRMSE = 0.5794163%

Figure 92: Actual and predicted engine deck responses with a Mach number: the worst NRMSE



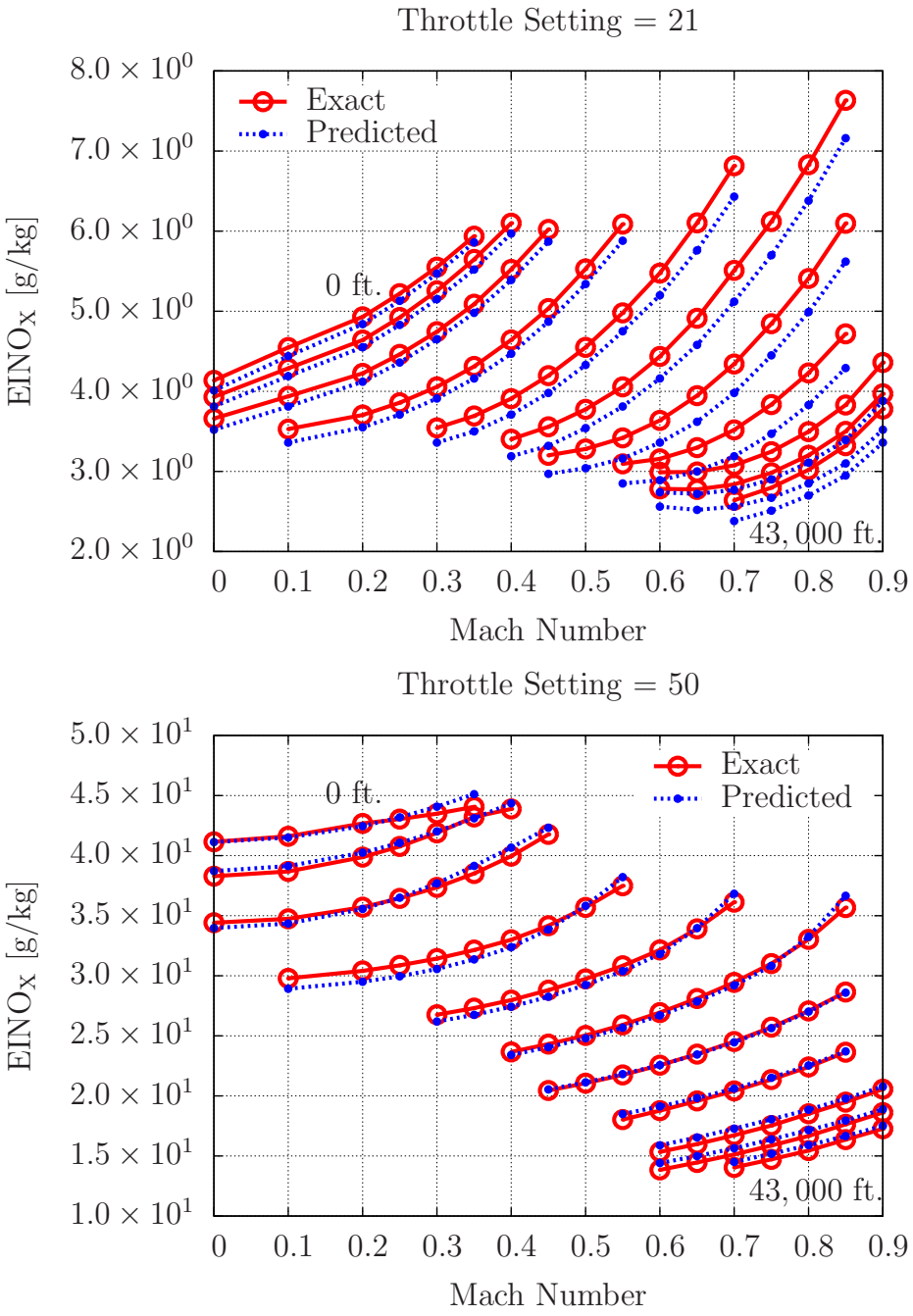
(d) Ram drag of the 289$^{\text{th}}$ test engine deck: NRMSE = 0.6556469%

Figure 92: Actual and predicted engine deck responses with a Mach number: the worst NRMSE



(f) Fuel flow of the 289$^{\text{th}}$ test engine deck: NRMSE = 1.441989%

Figure 92: Actual and predicted engine deck responses with a Mach number: the worst NRMSE



(h) $EINO_X$ of the 117^{th} test engine deck: NRMSE = 3.396025%

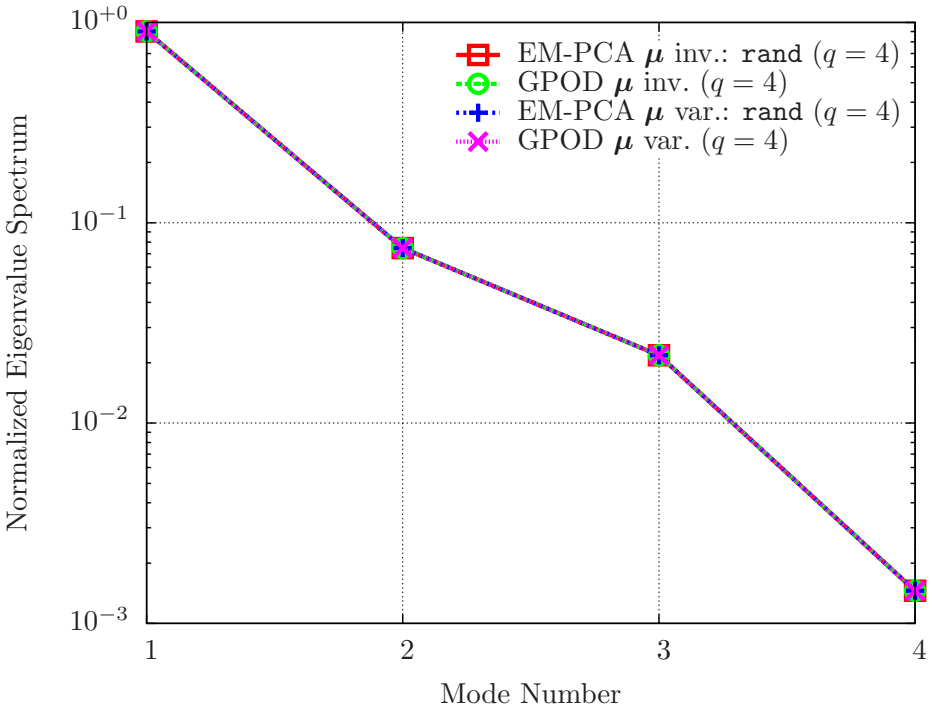
Figure 92: Actual and predicted engine deck responses with a Mach number: the worst NRMSE

B.3 Comparison of the Results of Gappy POD and the EM-PCA

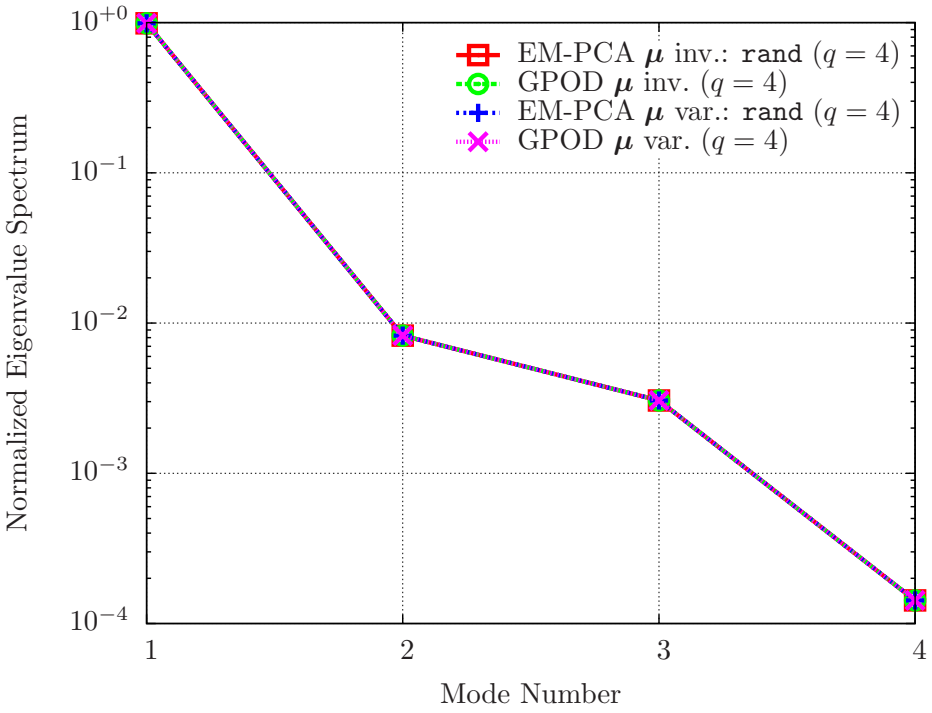
Although the validation of the EM-PCA against gappy POD is not of primary interest in Chapter 6, Lee et al.[34] compared the results of the EM-PCA with those of gappy POD to justify the use of the EM-PCA over gappy POD for computational efficiency. As similar to the notational convention for the EM-PCA implementations in Section 5.3.3.1, gappy POD implementations are denoted with "$\boldsymbol{\mu}$ inv."/"$\boldsymbol{\mu}$ var." to represent whether a sample mean is evaluated at each iteration. In addition, for only computational performance investigation, "Lanczos" is appended to the names of gappy POD implementations if the Lanczos algorithm is exploited to expedite a POD process. The Lanczos algorithm is realized in MATLAB with `eigs`, which invokes the Fortran Library ARPACK.[38]

First, the eigenvalues, eigenvectors, and restored failed performance analyses evaluated by the EM-PCA are compared to those evaluated by gappy POD in Figures 93, Figures 94–97, and Figures 98–99, respectively. The eigenspectra of the four engine deck responses in Figure 93 show that both the EM-PCA and gappy POD result in identical eigenvalues at the same number of modes. Similarly, the first four modes of the four engine deck responses, illustrated in Figures 94–97, also substantiate that the EM-PCA is capable of yielding the same POD bases as gappy POD. Last, as an example of restored failed performance analyses, both Figures 98 and 99 delineate the 101^{th} and 356^{th} training engine decks, respectively, both of which failed in a total of five off-design performance analyses. In Figures 98 and 99, all the estimated performance analyses are perfectly aligned regardless of the EM-PCA and gappy POD implementations. Note that failed analyses located at the highest and lowest throttle values are properly approximated as they follow the overall trend of engine deck responses because a POD basis can handle stationary discontinuities.

Last but not least, Figure 100 presents the results of computational performance tests measured in terms of computational time and the number of iterations with the four different snapshot ensembles of engine deck responses. For the minimal effect of random initialization on time measurements, the computational time of the EM-PCA implementations with `rand` was averaged over 100 runs. As shown in Figure 100, despite their higher numbers of iterations, the EM-PCA implementations are computationally more efficient than the gappy

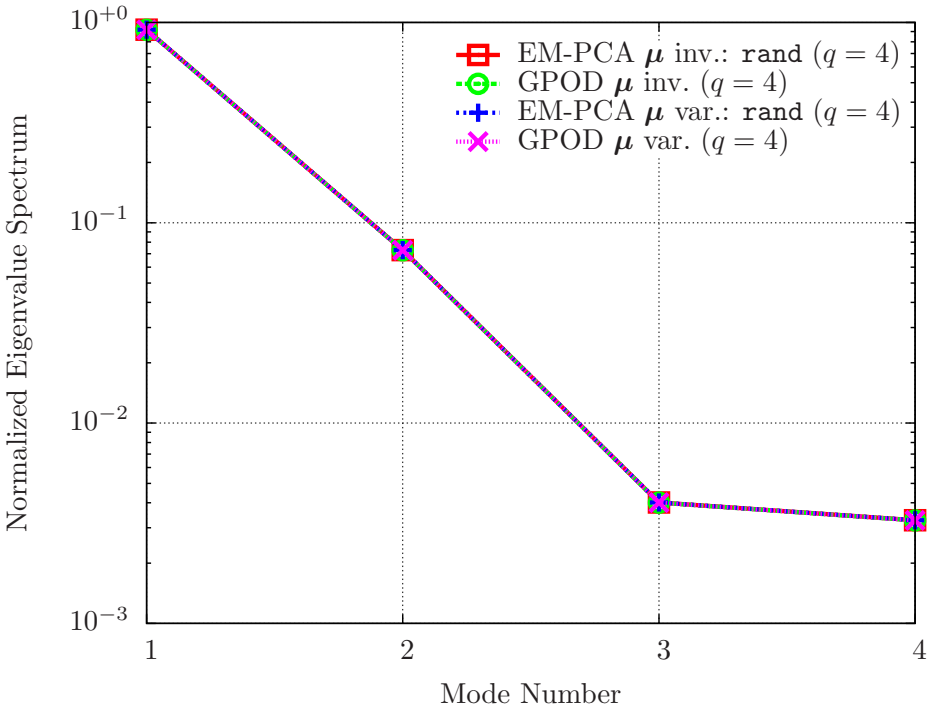


(a) Gross thrust

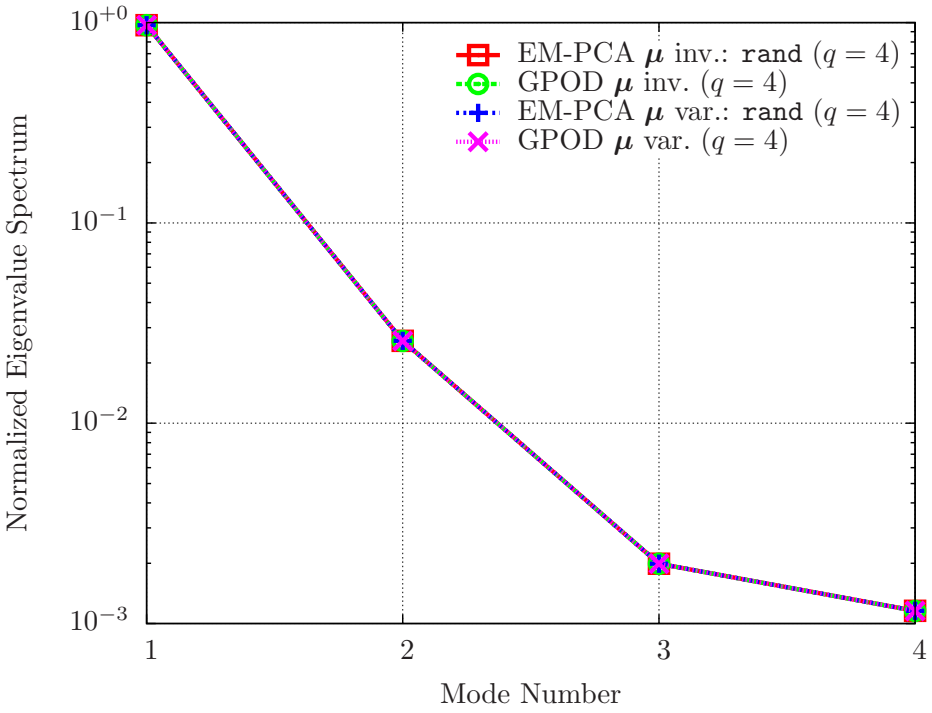


(b) Ram drag

Figure 93: Normalized eigenspectra of engine deck responses

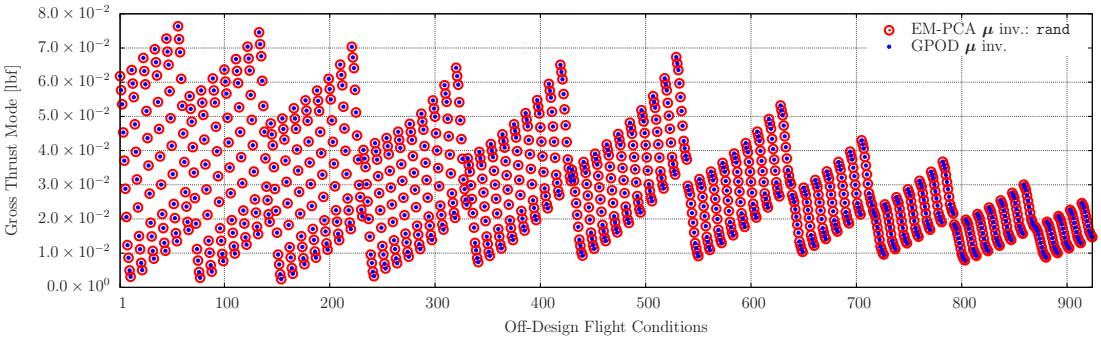


(c) Fuel flow

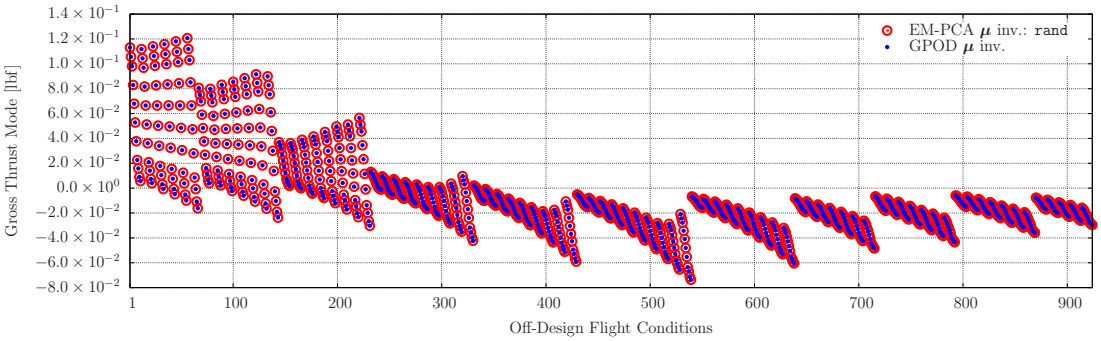


(d) EINO_X

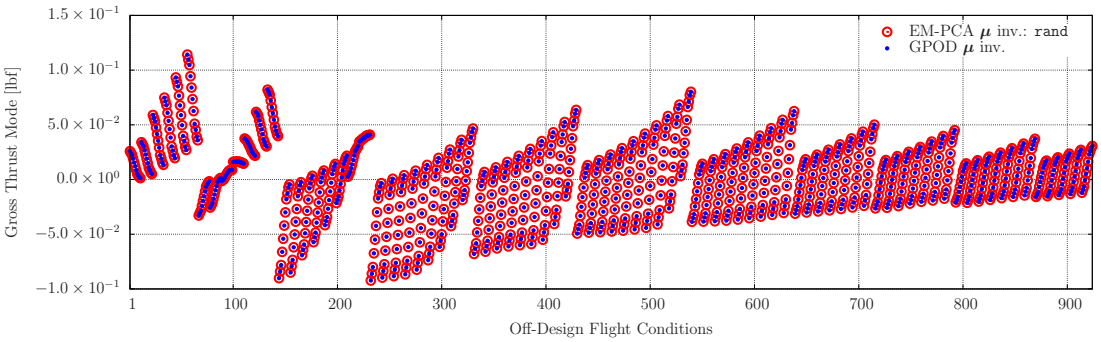
Figure 93: Normalized eigenspectra of engine deck responses



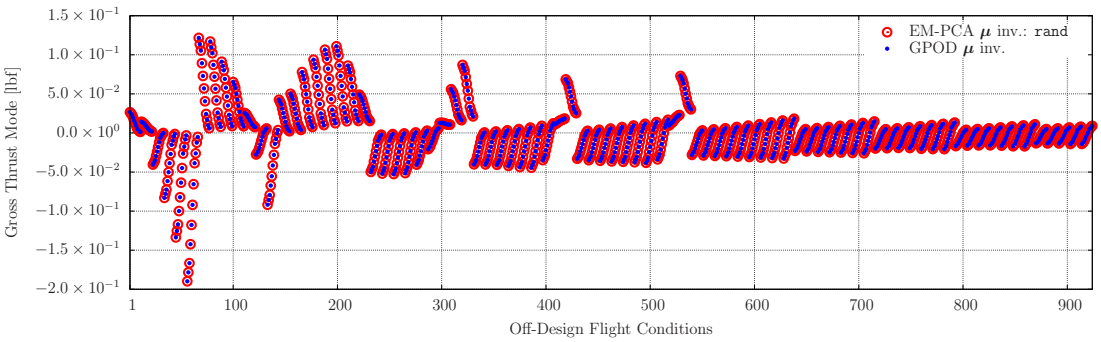
(a) 1st mode



(b) 2nd mode

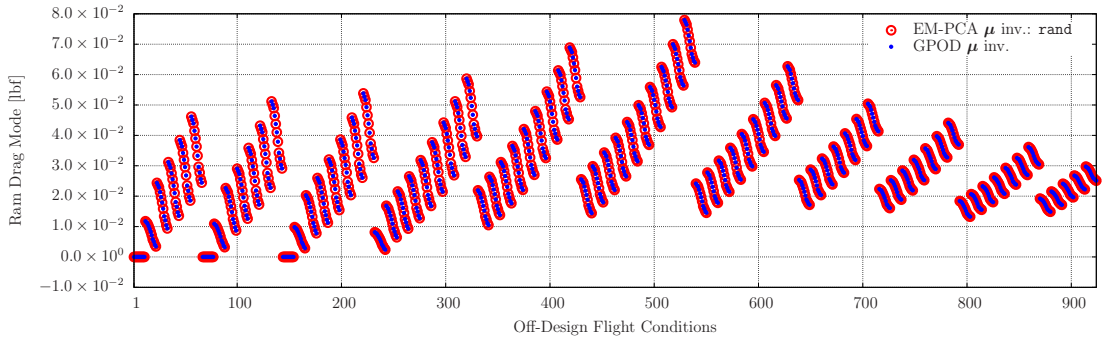


(c) 3rd mode

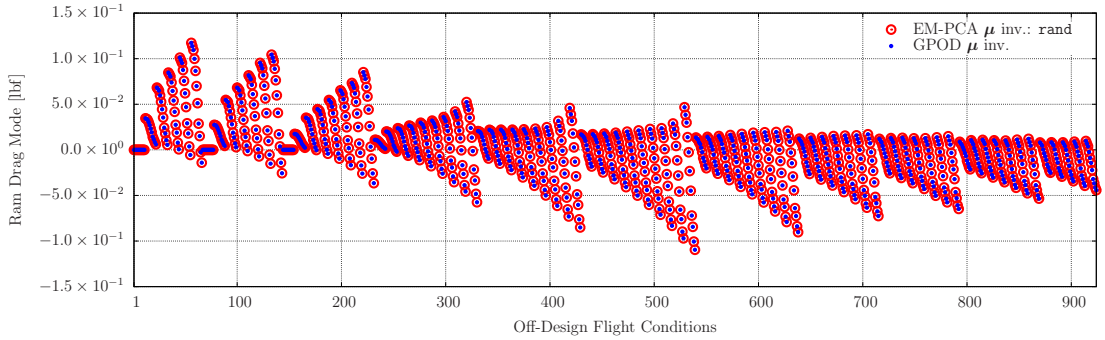


(d) 4th mode

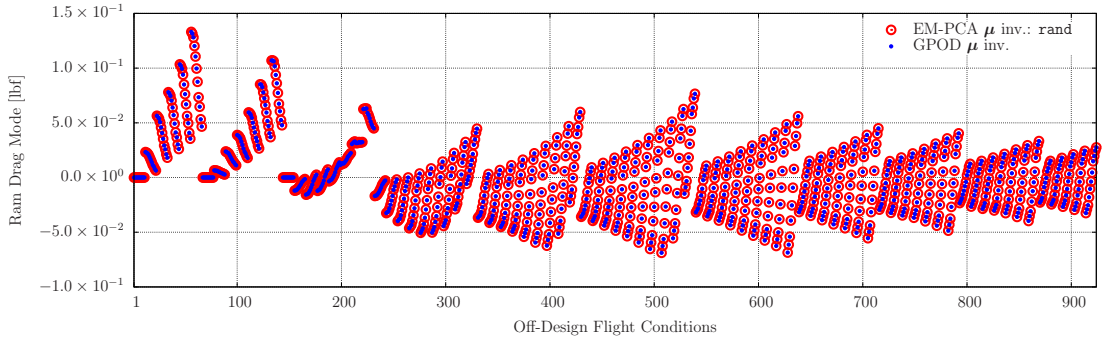
Figure 94: Modes of gross thrust evaluated by gappy POD and the EM-PCA



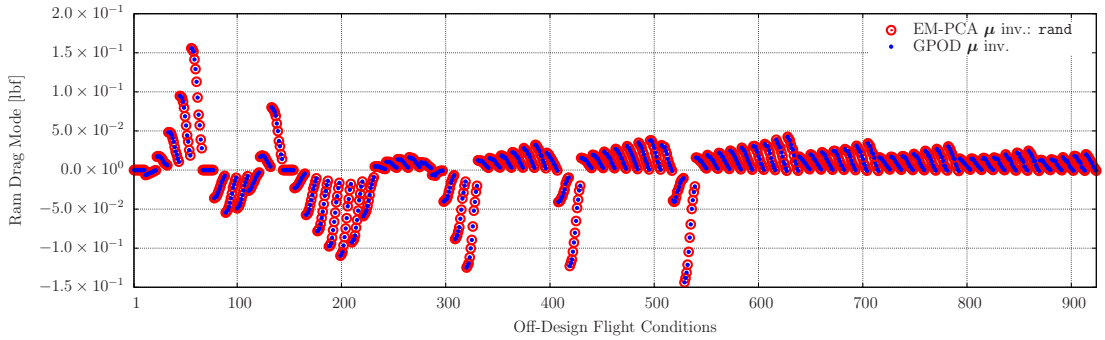
(a) 1st mode



(b) 2nd mode

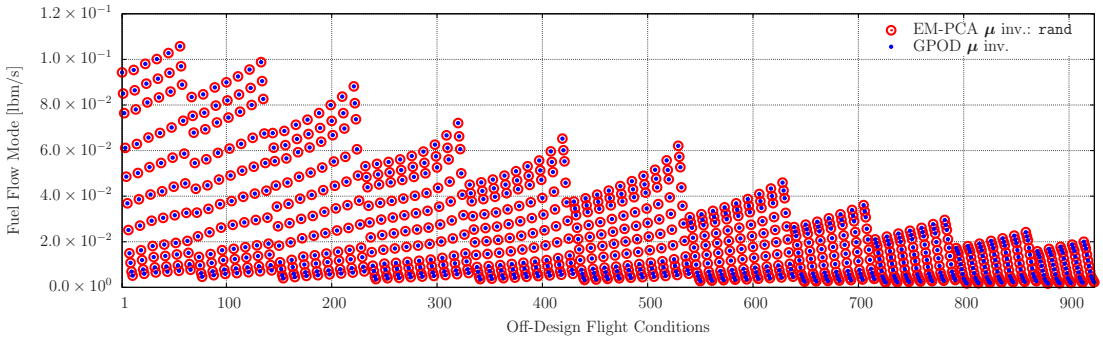


(c) 3rd mode

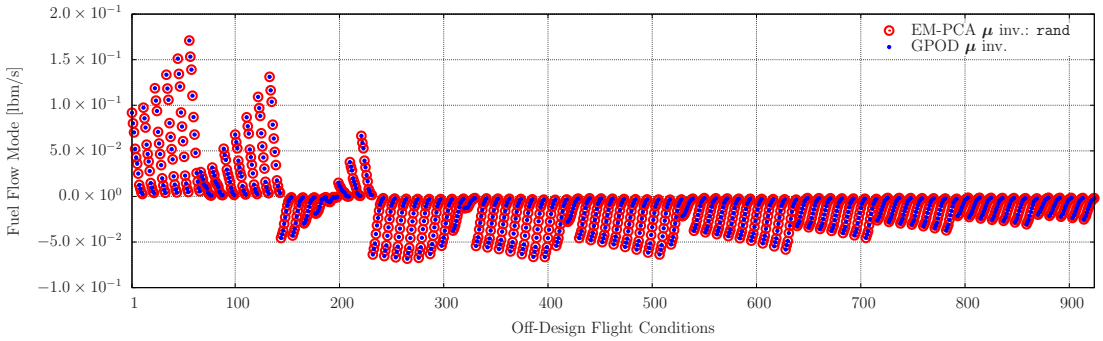


(d) 4th mode

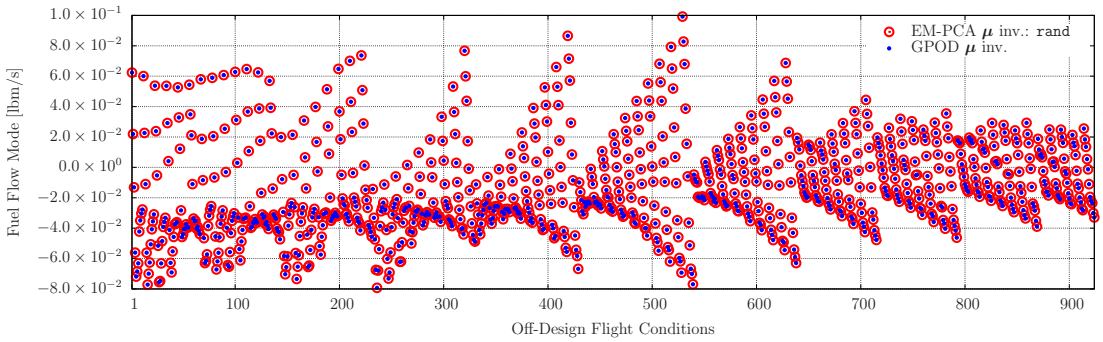
Figure 95: Modes of ram drag evaluated by gappy POD and the EM-PCA



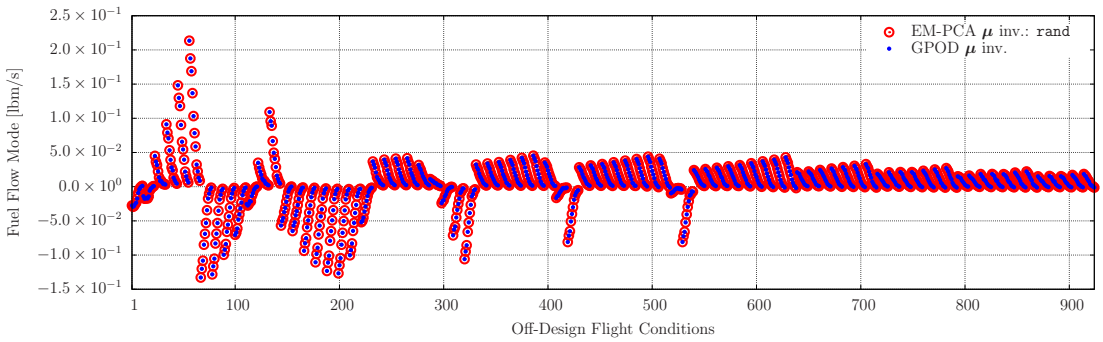
(a) 1st mode



(b) 2nd mode

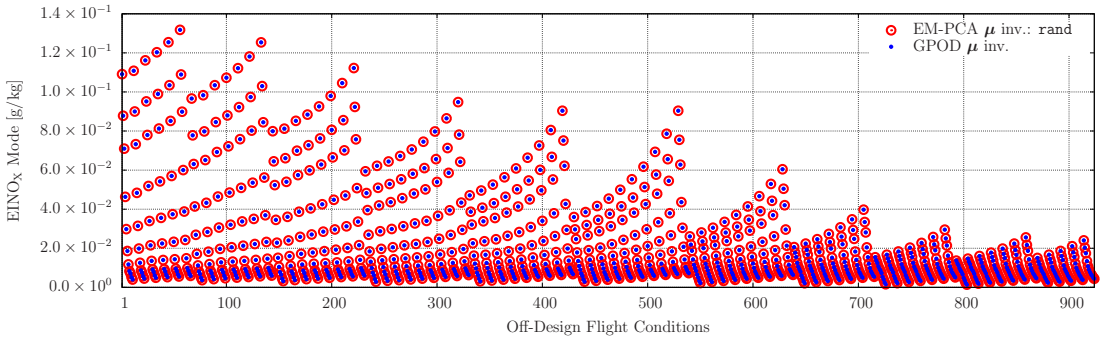


(c) 3rd mode

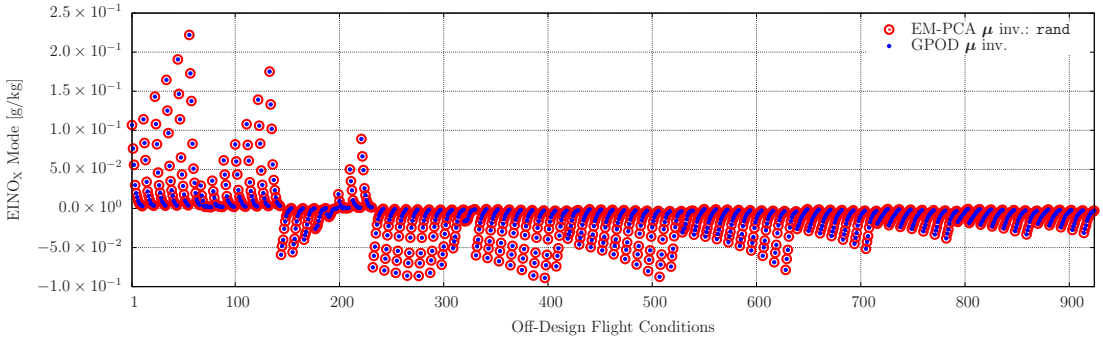


(d) 4th mode

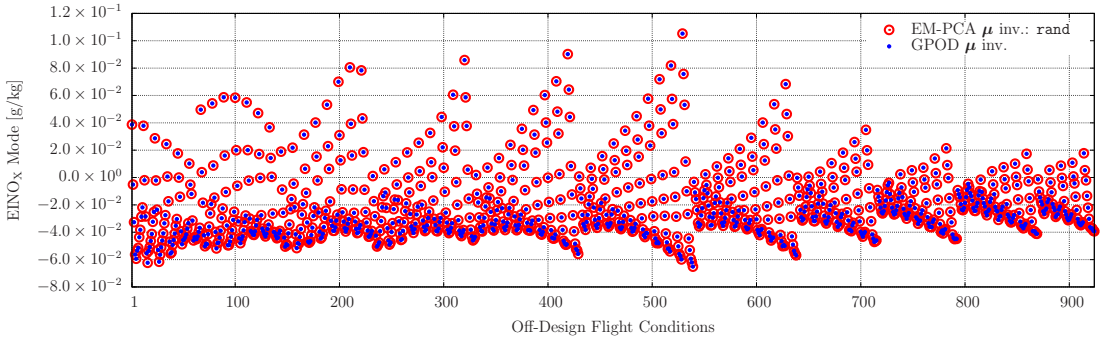
Figure 96: Modes of fuel flow evaluated by gappy POD and the EM-PCA



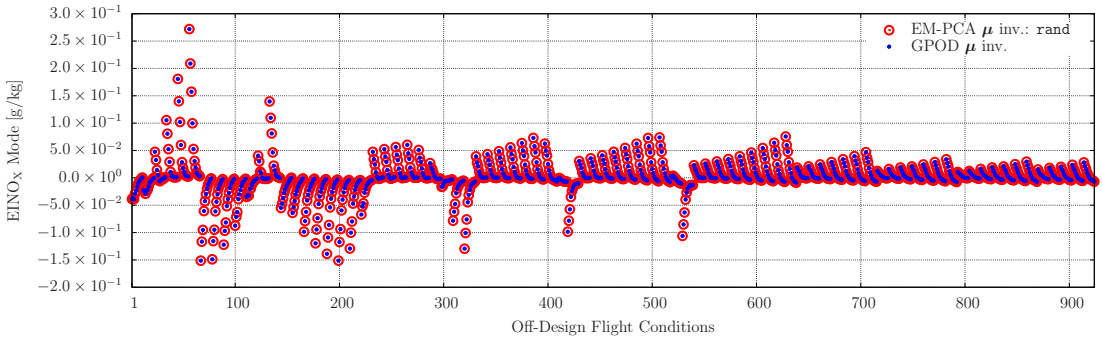
(a) 1st mode



(b) 2nd mode

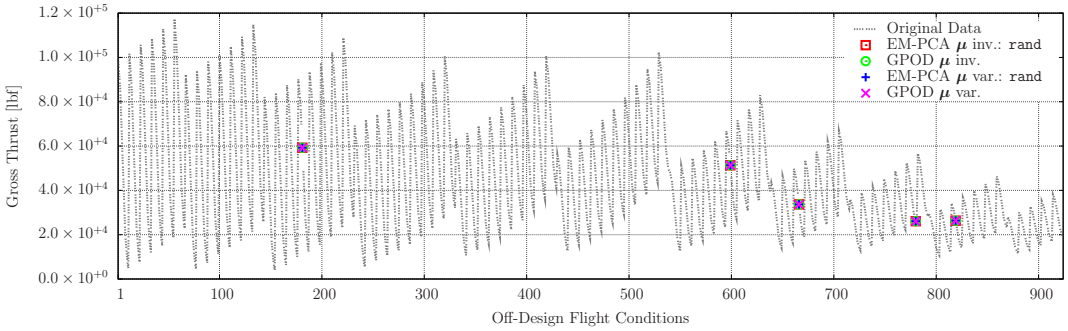


(c) 3rd mode

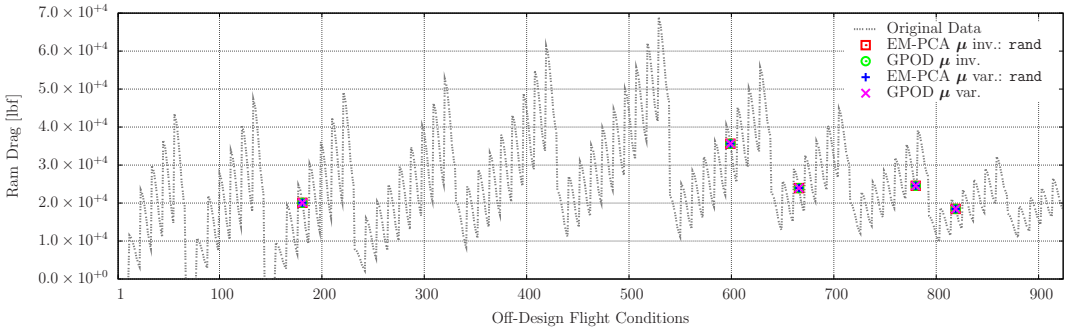


(d) 4th mode

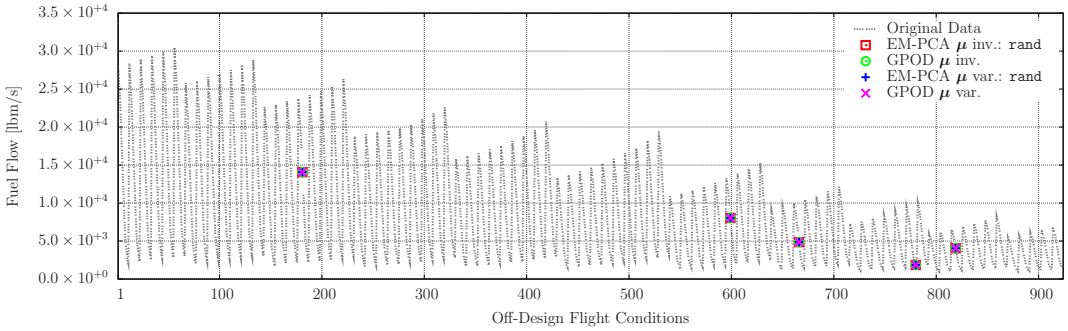
Figure 97: Modes of EINO$_X$ evaluated by gappy POD and the EM-PCA



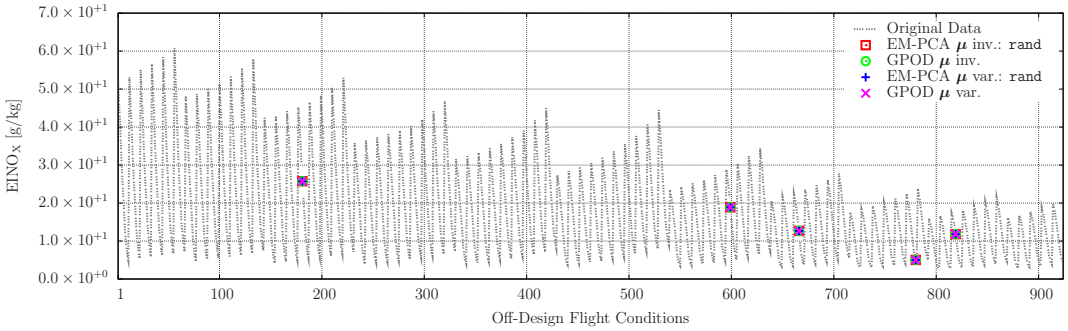
(a) Gross thrust



(b) Ram drag

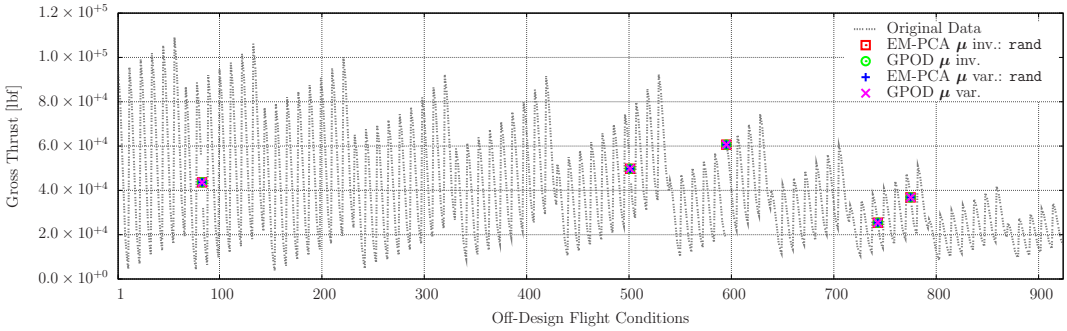


(c) Fuel flow

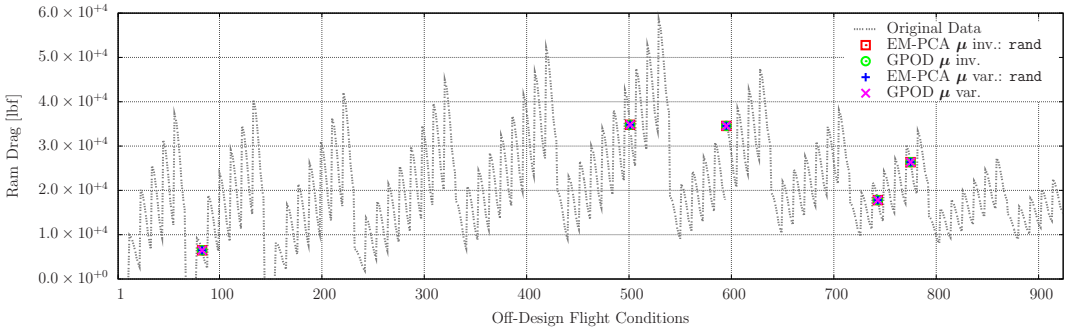


(d) $\mathrm{EINO_X}$

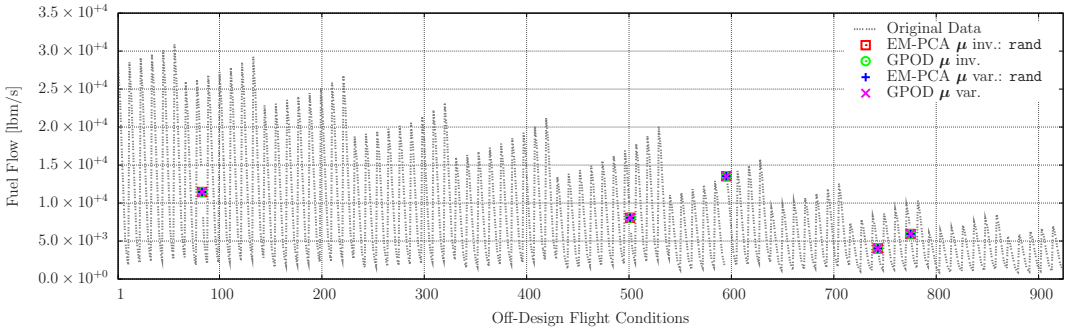
Figure 98: Restored engine deck responses: the 101^{th} training engine deck



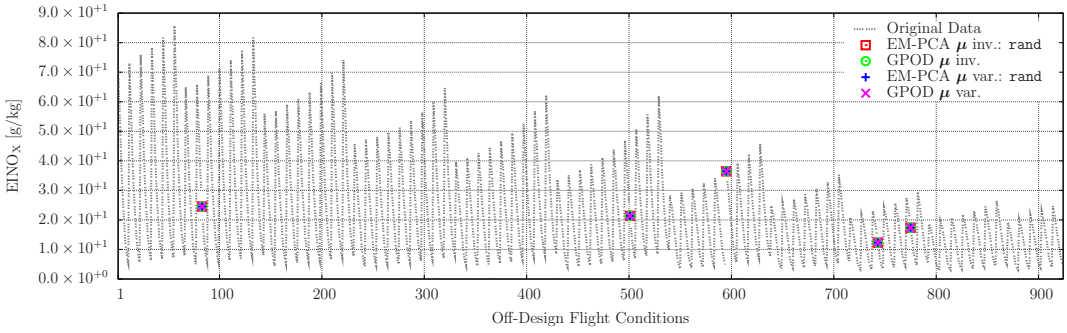
(a) Gross thrust



(b) Ram drag



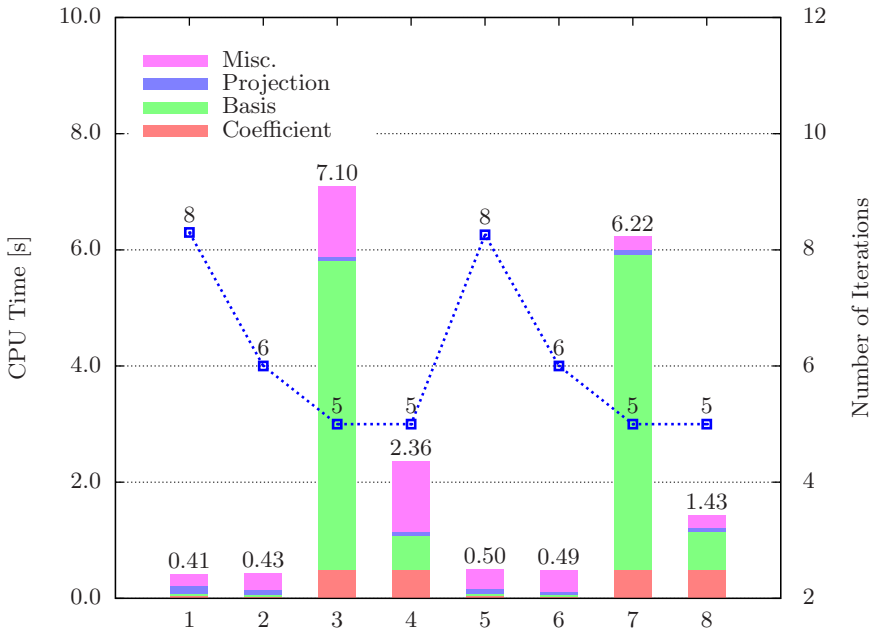
(c) Fuel flow



(d) $\mathrm{EINO_X}$

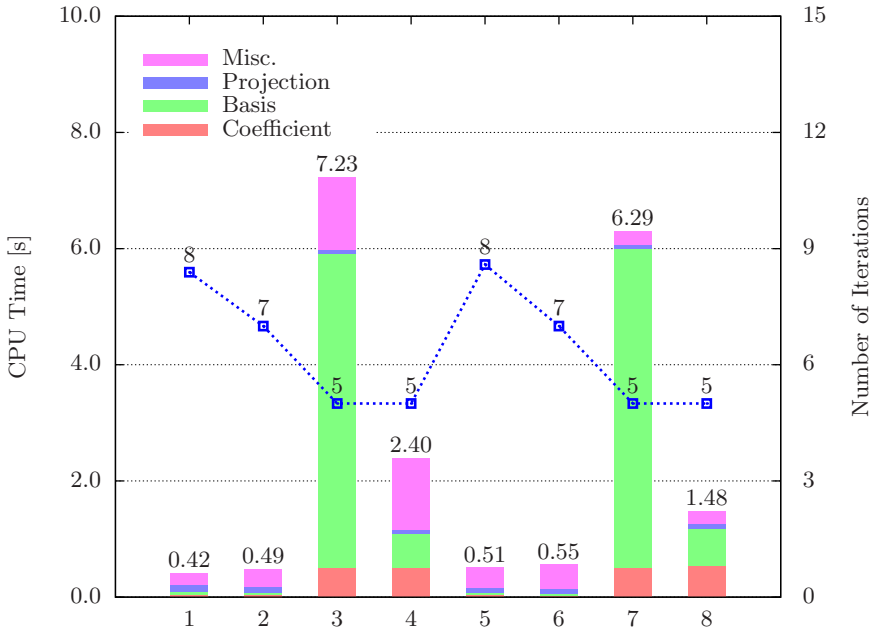
Figure 99: Restored engine deck responses: the 356^{th} training engine deck

POD implementations. The gappy POD implementations spent most of their computational time on basis evaluations since the size of a snapshot ensemble cubically affects the POD process. Although the Lanczos algorithm is particularly conducive to accelerating basis evaluations (e.g., a situation in which the number of modes to extract is small such as four), it was not effective enough for gappy POD implementations to outperform the EM-PCA implementations. Note that $\mathbf{V}_e$ for the EM-PCA implementations was also evaluated with the Lanczos algorithm for computational efficiency due to the large number of snapshots, 500. After all, even though the EM-PCA necessitates an extra step for the orthogonalization of its non-orthogonal basis, it generally surpasses gappy POD for the given snapshots of engine deck responses, characterized by the absence of randomly-scattered data.



1: EM-PCA $\boldsymbol{\mu}$ inv.: `rand` ($q = 4$)
2: EM-PCA $\boldsymbol{\mu}$ inv.: $\mathbf{V}_e$ ($q = 4$)
3: GPOD $\boldsymbol{\mu}$ inv. ($q = 4$)
4: GPOD $\boldsymbol{\mu}$ inv.: Lanczos ($q = 4$)
5: EM-PCA $\boldsymbol{\mu}$ var.: `rand` ($q = 4$)
6: EM-PCA $\boldsymbol{\mu}$ var.: $\mathbf{V}_e$ ($q = 4$)
7: GPOD $\boldsymbol{\mu}$ var. ($q = 4$)
8: GPOD $\boldsymbol{\mu}$ var.: Lanczos ($q = 4$)

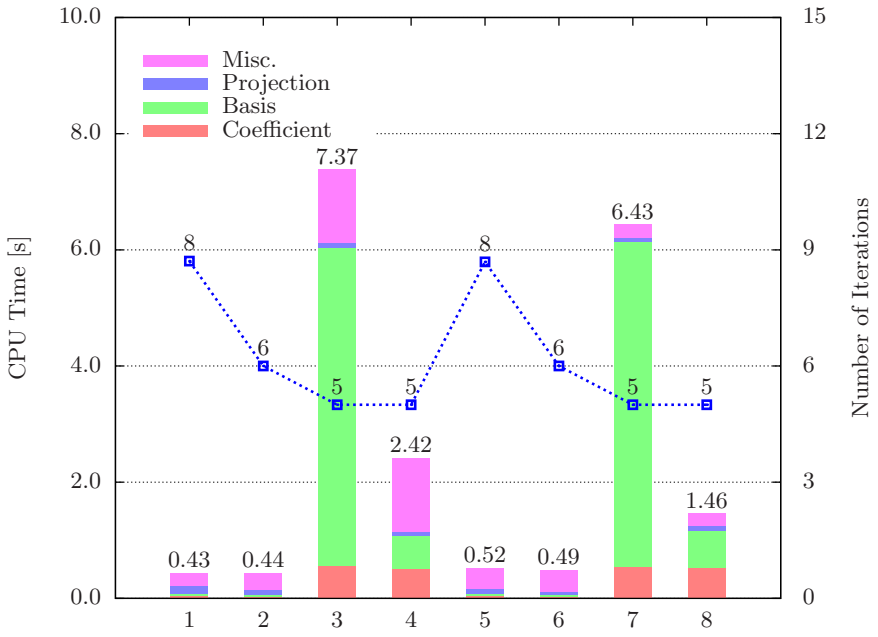
(a) Gross thrust



1: EM-PCA $\boldsymbol{\mu}$ inv.: `rand` ($q = 4$)
2: EM-PCA $\boldsymbol{\mu}$ inv.: $\mathbf{V}_e$ ($q = 4$)
3: GPOD $\boldsymbol{\mu}$ inv. ($q = 4$)
4: GPOD $\boldsymbol{\mu}$ inv.: Lanczos ($q = 4$)
5: EM-PCA $\boldsymbol{\mu}$ var.: `rand` ($q = 4$)
6: EM-PCA $\boldsymbol{\mu}$ var.: $\mathbf{V}_e$ ($q = 4$)
7: GPOD $\boldsymbol{\mu}$ var. ($q = 4$)
8: GPOD $\boldsymbol{\mu}$ var.: Lanczos ($q = 4$)

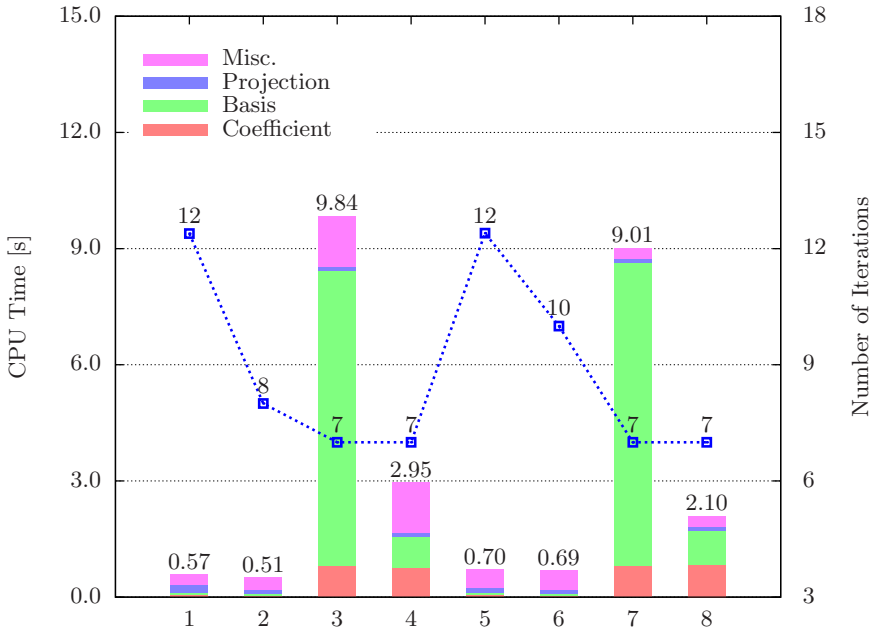
(b) Ram drag

Figure 100: Computational time decomposition along with numbers of iterations



1: EM-PCA $\boldsymbol{\mu}$ inv.: `rand` ($q = 4$)
2: EM-PCA $\boldsymbol{\mu}$ inv.: $\mathbf{V}_e$ ($q = 4$)
3: GPOD $\boldsymbol{\mu}$ inv. ($q = 4$)
4: GPOD $\boldsymbol{\mu}$ inv.: Lanczos ($q = 4$)
5: EM-PCA $\boldsymbol{\mu}$ var.: `rand` ($q = 4$)
6: EM-PCA $\boldsymbol{\mu}$ var.: $\mathbf{V}_e$ ($q = 4$)
7: GPOD $\boldsymbol{\mu}$ var. ($q = 4$)
8: GPOD $\boldsymbol{\mu}$ var.: Lanczos ($q = 4$)

(c) Fuel flow



1: EM-PCA $\boldsymbol{\mu}$ inv.: `rand` ($q = 4$)
2: EM-PCA $\boldsymbol{\mu}$ inv.: $\mathbf{V}_e$ ($q = 4$)
3: GPOD $\boldsymbol{\mu}$ inv. ($q = 4$)
4: GPOD $\boldsymbol{\mu}$ inv.: Lanczos ($q = 4$)
5: EM-PCA $\boldsymbol{\mu}$ var.: `rand` ($q = 4$)
6: EM-PCA $\boldsymbol{\mu}$ var.: $\mathbf{V}_e$ ($q = 4$)
7: GPOD $\boldsymbol{\mu}$ var. ($q = 4$)
8: GPOD $\boldsymbol{\mu}$ var.: Lanczos ($q = 4$)

(d) EINO_X

Figure 100: Computational time decomposition along with numbers of iterations

Bibliography

[1] Acharjee, S. and Zabaras, N., "A concurrent model reduction approach on spatial and random domains for the solution of stochastic PDEs," *International Journal for Numerical Methods in Engineering*, vol. 66, pp. 1934–1954, Jananuary 13 2006.

[2] Alfeld, P., "Scattered data interpolation in three or more variables," in *Mathematical Methods in Computer Aided Geometric Design* (Lyche, T. and Schumaker, L., eds.), pp. 1–34, Academic Press, 1989.

[3] Bauchau, O. and Craig, J., *Structural Analysis: With Applications to Aerospace Structures*. Solid Mechanics and Its Applications, Springer, September 21 2009.

[4] Bui-Thanh, T., "Proper orthogonal decomposition extensions and their applications in steady aerodynamics," Master's thesis, Department of Aeronautics and Astronautics, M.I.T., 2003.

[5] Bui-Thanh, T., Damodaran, M., and Willcox, K., "Proper orthogonal decomposition extensions for parametric applications in compressible aerodynamics," in *21st AIAA Applied Aerodynamics Conference*, no. AIAA-2003-4213, (Orlando, Florida), June 23–26 2003.

[6] Bui-Thanh, T., Damodaran, M., and Willcox, K., "Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition," *AIAA Journal*, vol. 42, pp. 1505–1516, August 2004.

[7] Cazemier, W., *Proper orthogonal decomposition and low dimensional models for turbulent flows*. PhD thesis, University of Groningen, September 1997.

[8] Chen, T., Martin, E., and Montague, G., "Robust probabilistic pca with missing data and contribution analysis for outlier detection," *Computational Statistics & Data Analysis*, vol. 53, pp. 3706–3716, August 2009.

[9] Debnath, L. and Mikusinski, P., *Introduction to Hilbert Spaces with Applications*. Academic Press, 3 edition ed., September 29 2005.

[10] Dempster, A. P., Laird, N. M., and Rubin, D. B., "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[11] Drela, M., "Transport aircraft optimization." Georgia Institute of Technology, December 3 2009. Aerospace Engineering Fall 2009 Seminar Series.

[12] Du, K. L. and Swamy, M. N. S., *Neural Networks in a Softcomputing Framework*. Springer, 2006.

[13] Everson, R. and Sirovich, L., "Karhunen-Loève procedure for gappy data," *Journal of the Optical Society of America A: Optics and Image Science, and Vision*, vol. 12, pp. 1657–1664, August 1995.

[14] Fujino, M., Yoshizaki, Y., and Kawamura, Y., "Natural-laminar-flow airfoil development for a lightweight business jet," *Journal of Aircraft*, vol. 40, pp. 609–615, July-August 2003.

[15] Golub, G. H. and van Loan, C. F., *Matrix Computations*. Johns Hopkins University Press, 2d ed. ed., 1989.

[16] Gunes, H., Sirisup, S., and Karniadakis, G. E., "Gappy data: To krig or not to krig?," *Journal of Computational Physics*, vol. 212, pp. 358–382, February 10 2006. Kriging; proper orthogonal decomposition; unsteady flow.

[17] Hodges, D. H. and Pierce, G. A., *Introduction to Structural Dynamics and Aeroelasticity*. Cambridge University Press, July 1 2002.

[18] Holmes, P., Lumley, J. L., and Berkooz, G., *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge University Press, August 13 1998. ISBN-10: 0521634199.

[19] Houseago-Stokes, R. E. and Challenor, P. G., "Using PPCA to estimate EOFs in the presence of missing values," *Journal of Atmospheric and Oceanic Technology*, vol. 21, pp. 1471–1480, September 1 2004.

[20] Huang, H.-S., Yang, B.-H., and Hsu, C.-N., "Triple jump acceleration for the em algorithm," *Data Mining, IEEE International Conference on*, vol. 0, pp. 649–652, 2005.

[21] Hwang, D. and Zeng, G. L., "Convergence study of an accelerated ML-EM algorithm using bigger step size," *Physics in Medicine and Biology*, vol. 51, no. 2, pp. 237–252, 2006.

[22] J. Zhao, Q. J., "Probabilistic PCA for t distributions," *Neurocomputing*, vol. 69, no. 16-18, pp. 2217–2226, 2006. Brain Inspired Cognitive Systems - Selected papers from the 1st International Conference on Brain Inspired Cognitive Systems (BICS 2004).

[23] Jameson, A., "Iterative solution of transonic flows over airfoils and wings, including flows at mach 1," *Communications on Pure and Applied Mathematics*, vol. 27, pp. 283–309, May 1974.

[24] JMP, *JMP Design of Experiments*. SAS Institute Incorporated, Cary, NC, USA, release 6 ed., 2005.

[25] Jolliffe, I., *Principal Component Analysis*. Springer Series in Statistics, Springer, 2nd ed., October 1 2002. ISBN: 978-0-387-95442-4.

[26] Jones, S. M., "An introduction to thermodynamic performance analysis of aircraft gas turbine engine cycles using the numerical propulsion system simulation code," Tech. Rep. NASA/TM-2007-214690, NASA, Glenn Research Center, Cleveland, Ohio 44135, March 2007.

[27] Kaufman, L., "Implementing and accelerating the EM algorithm for positron emission tomography.," *IEEE Transactions on Medical Imaging*, vol. MI-6, no. 1, pp. 37–51, 1987. EMISSION TOMOGRAPHY;POSITRON EMISSION TOMOGRAPHY;.

[28] Lavelle, T. M., Plencner, R. M., and Seidel, J. A., "Concurrent optimization of airframe and engine design parameters," Tech. Rep. NASA TM-105908, NASA, Glenn Research Center, Cleveland, Ohio 44135, September 1 1992. AIAA-1992-4713, Prepared for the Fourth Symposium on Multidisciplinary Analysis and Optimization cosponsored by the AIAA, USAF, NASA, and OAI, September 21-23, 1992.

[29] Lawrence, N., "Probabilistic non-linear principal component analysis with gaussian process latent variable models," *Journal of Machine Learning Research*, vol. 6, pp. 1783–1816, December 2005.

[30] Lee, K. and Mavris, D. N., "Comparison of gappy proper orthogonal decomposition and probabilistic principal component analysis in terms of their basis and norm differences," *Journal of Computational Physics*. submitted in March 2010.

[31] Lee, K. and Mavris, D. N., "Efficient piv data restoration via probabilistic principal component analysis," *Experiments in fluids*. submitted in December 2009.

[32] Lee, K. and Mavris, D. N., "A unifying least squares perspective for gappy proper orthogonal decomposition and probabilistic principal component analysis," in *39th AIAA Fluid Dynamics Conference*, no. AIAA-2009-3899, (San Antonio, Texas), June 23 2009.

[33] Lee, K. and Mavris, D. N., "Unifying perspective for gappy proper orthogonal decomposition and probabilistic principal component analysis," *AIAA Journal*, vol. 48, pp. 1117–1129, June 2010. submitted in June 2009, accepted in January 2010.

[34] Lee, K., Nam, T., Perullo, C., and Mavris, D. N., "Reduced-order modeling of a high-fidelity propulsion system simulation via probabilistic principal component analysis and neural networks," in *13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, (Fort Worth, Texas), September 13 - 15 2010. abstract accepted.

[35] Lee, K., Rallabhandi, S. K., and Mavris, D. N., "Aerodynamic data reconstruction via probabilistic principal component analysis," in *46th AIAA Aerospace Sciences Meeting and Exhibit*, no. AIAA-2008-899, (Reno, Nevada), January 7–10 2008.

[36] LeGresley, P. A., *Application of Proper Orthogonal Decomposition (POD) to Design Decomposition Methods*. PhD thesis, Stanford University, October 2005.

[37] LeGresley, P. A. and Alonso, J. J., "Airfoil design optimization using reduced order models based on proper orthogonal decomposition," in *FLUIDS 2000 Conference and Exhibit*, no. AIAA 2000-2545, (Denver, CO), June 19–22 2000.

[38] Lehoucq, R. B., Sorensen, D. C., and Yang, C., *ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*. SIAM Publications, Philadelphia, 1998.

[39] Lucia, D. J., *Reduced order modeling for high speed flows with moving shocks*. PhD thesis, Air Force Institute of Technology, November 2001.

[40] Lucia, D. J., Beran, P. S., and Silva, W. A., "Reduced-order modeling: New approaches for computational physics," *Progress in Aerospace Sciences*, vol. 40, pp. 51–117, February 2004.

[41] Lucia, D. J., King, P. I., and Beran, P. S., "Domain decomposition for reduced-order modeling of a flow with moving shocks," *AIAA Journal*, vol. 40, pp. 2360–2362, November 2002.

[42] Lucia, D. J., King, P. I., and Beran, P. S., "Reduced order modeling of a two-dimensional flow with moving shocks," *Computers & Fluids*, vol. 32, pp. 917–938, August 2003.

[43] Ly, H. V. and Tran, H. T., "Modeling and control of physical processes using proper orthogonal decomposition," *Mathematical and Computer Modelling*, vol. 33, no. 1–3, pp. 223–236, 2001. Computation and control VI proceedings of the sixth Bozeman conference.

[44] Lytle, J. K., "The numerical propulsion system simulation: An overview," Tech. Rep. NASA/TM-2000-209915, NASA, Glenn Research Center, Cleveland, Ohio 44135, June 2000.

[45] Mahadevan, S., "Fast spectral learning using Lanczos eigenspace projections," in *AAAI'08: Proceedings of the 23rd national conference on Artificial intelligence*, pp. 1472–1475, AAAI Press, 2008.

[46] Malone, J. B. and Sankar, L., "Numerical simulation of two-dimensional unsteady transonic flows using the full potential equation," *AIAA Journal*, vol. 22, pp. 1035–1041, August 1984.

[47] Mattingly, J. D., Heiser, W. H., and Pratt, D. T., *Aircraft Engine Design*. AIAA Education Series, AIAA, December 2002.

[48] Mavris, D. N., Bandte, O., and DeLaurentis, D. A., "Robust design simulation: A probabilistic approach to multidisciplinary design," *Journal of Aircraft*, vol. 36, pp. 298–307, January–February 1999.

[49] Mavris, D. N. and DeLaurentis, D. A., "A probabilistic approach for examining aircraft concept feasibility and viability," *Aircraft Design*, vol. 3, pp. 79–101, June 2000.

[50] McGregor, R., Szczerba, D., von Siebenthal, M., Muralidhar, K., and Székely, G., *Medical Image Computing and Computer-Assisted Intervention MICCAI 2008*, vol. 5242/2008, ch. Exploring the Use of Proper Orthogonal Decomposition for Enhancing Blood Flow Images Via Computational Fluid Dynamics, pp. 782–789. Springer Berlin/Heidelberg, 2008.

[51] Michael Collins, Sanjoy Dasgupta, R. E. S., "A generalization of principal component analysis to the exponential family," *Advances In Neural Information Processing Systems*, vol. 1, no. 14, pp. 617–624, 2002.

[52] Mifsud, M. J., Shaw, S. T., and MacManus, D. G., "A high-fidelity low-cost aerodynamic model using proper orthogonal decomposition," *International Journal for Numerical Methods in Fluids*, 2009. Published Online on June 1, 2009.

[53] Mifsud, M., Shaw, S., and Bennett, J., "A meta-modeling technique using POD in parametric studies of weapon aerodynamics," in *AIAA Atmospheric Flight Mechanics*

Conference and Exhibit, no. AIAA 2006-6005, (Keystone, Colorado), August 21–24 2006.

[54] Min, B. Y., Lee, W., Englar, R., and Sankar, L. N., "Numerical investigation of circulation control airfoils," in *46th AIAA Aerospace Sciences Meeting and Exhibit*, no. AIAA 2008-0329, (Reno, Nevada), January 2008.

[55] Min, B. Y., *A Physics Based Investigation of Gurney Flaps for Enhancement of Rotorcraft Flight Characteristics*. PhD thesis, School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, May 2010.

[56] Min, B. Y., Sankar, L. N., Rajmohan, N., and Prasad, J., "Computational investigation of the effects of gurney flaps on rotors in forward flight," *Journal of Aircraft*, vol. 46, pp. 1957–1964, November–December 2009.

[57] Murray, N. E. and Seiner, J. M., "The effects of gappy POD on higher-order turbulence quantities," in *46th AIAA Aerospace Sciences Meeting and Exhibit*, no. AIAA 2008-241, (Reno, Nevada), January 7–10 2008.

[58] Murray, N. E. and Ukeiley, L. S., "An application of gappy POD: For subsonic cavity flow PIV data," *Experiments in Fluids*, vol. 42, pp. 79–91, January 2007.

[59] Petersen, K. B. and Winther, O., "Explaining slow convergence of EM in low noise linear mixtures," tech. rep., Informatics and Mathematical Modelling, Technical University of Denmark, DTU, 2005.

[60] Petersen, K. B., Winther, O., and Hansen, L. K., "On the slow convergence of EM and VBEM in low-noise linear models," *Neural Computation*, vol. 17, no. 9, pp. 1921–1926, 2005.

[61] Raffel, M., Willert, C. E., Wereley, S. T., and Kompenhans, J., *Particle Image Velocimetry*, ch. Post-Processing of PIV Data, pp. 177–208. Experimental Fluid Mechanics, Springer Berlin Heidelberg, second edition ed., Friday, September 14 2007.

[62] Robinson, T. D., Eldred, M. S., Willcox, K. E., and Haimes, R., "Strategies for multifidelity optimization with variable dimensional hierarchical models," in *47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, no. AIAA-2006-1819, (7th, Newport, Rhode Island), May 1–4 2006.

[63] Roweis, S., "EM algorithms for PCA and SPCA," in *in Advances in Neural Information Processing Systems*, pp. 626–632, MIT Press, 1998.

[64] Rubin, D. B. and Thayer, D. T., "EM algorithms for ML factor analysis," *Psychometrika*, vol. 47, pp. 69–76, March 1982.

[65] Sacks, J., Welch, W., Mitchell, T., and Wynn, H., "Design and analysis of computer experiments," *Statistical Science*, vol. 4, pp. 409–423, November 1989.

[66] SAE International, *Gas turbine engine steady state and transient performance presentation for digital computer programs*, 1999.

[67] Salakhutdinov, R. and Roweis, S., "Adaptive overrelaxed bound optimization methods," vol. 2, (Washington, DC, United States), pp. 664–671, 2003. Iterative scaling (IS);Non-negative matrix factorization;.

[68] Santner, T. J., Williams, B. J., and Notz, W., *The Design and Analysis of Computer Experiments*. Springer Series in Statistics, Springer, 1 edition ed., 2003.

[69] Schobeiri, M., *Turbomachinery Flow Physics and Dynamic Performance*. New York: Springer, December 2004.

[70] Schuster, M., Hori, T., and Nakamura, A., "Experiments with probabilistic principal component analysis in LVCSR," in *Interspeech'2005 (Eurospeech)*, (Lisbon, Portugal), pp. 1685–1688, 2005. Acoustic model;Error rates;Mixtures of Probabilistic Principal Component Analyzers (MPPCA);Observation distributions;.

[71] Senzig, D. A., Fleming, G. G., and Iovinelli, R. J., "Modeling of terminal-area airplane fuel consumption," *Journal of Aircraft*, vol. 46, pp. 1089–1093, July-August 2009.

[72] Shanbhogue, S., Shin, D.-H., Hemchandra, S., Plaks, D., and Lieuwen, T., "Flame sheet dynamics of bluff-body stabilized flames during longitudinal acoustic forcing," in *Proceedings of the Combustion Institute*, vol. 32, pp. 1787–1794, 2009.

[73] Shanbhogue, S. J., *Dynamics of perturbed exothermic bluff-body flow-fields*. PhD thesis, Georgia Institute of Technology, July 8 2008.

[74] Shlens, J., "A tutorial on principal component analysis." Version 2, December 10 2005.

[75] Simpson, T. W., Lin, D. K. J., and Chen, W., "Sampling strategies for computer experiments: Design and analysis," *International Journal of Reliability and Applications*.

[76] Simpson, T. W., Poplinski, J. D., Koch, P. N., and Allen, J. K., "Metamodels for computer-based engineering design: Survey and recommendations.," *Engineering with Computers*, vol. 17, pp. 129–150, July 2001.

[77] Sirovich, L., "Turbulence and the dynamics of coherent structures," *Quarterly of Applied Mathematics*, vol. 45, pp. 561–571, 573–590, October 1987.

[78] Sjöberg, J., *Mathematica Neural Networks: Train and Analyze Neural Networks to Fit Your Data*. Wolfram Research, Inc., first ed., September 2005.

[79] Tang, K., Graham, W., and Peraire, J., "Active flow control using a reduced order model and optimum control," in *27th Fluid Dynamics Conference*, no. AIAA 1996-1946, (New Orleans, LA), June 17–20 1996.

[80] Thomas, J. P., Hall, K. C., and Dowell, E. H., "Reduced-order aeroelastic modeling using proper-orthogonal decompostions," in *CEAS/AIAA/ICASE/NASA Langley International Forum on Aeroelasticity and Structural Dynamics*, (Williamsburg, Virginia), June 1999.

[81] Tipping, M. E. and Bishop, C. M., "Mixtures of probabilistic principal component analyzers," *Neural Computation*, vol. 11, pp. 443–482, February 15 1999.

[82] Tipping, M. E. and Bishop, C. M., "Probabilistic principal component analysis," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 61, no. 3, pp. 611–622, 1999.

[83] van der Maaten, L., Postma, E., and van den Herik, H., "Dimensionality reduction: A comparative review." May 7 2007.

[84] Venturi, D. and Karniadakis, G. E., "Gappy data and reconstruction procedures for flow past a cylinder," *Journal of Fluid Mechanics*, vol. 519, pp. 315–336, 2004.

[85] Veres, J. P., "Overview of high-fidelity modeling activities in the numerical propulsion system simulations (NPSS) project," Tech. Rep. NASA/TM-2002-211351, NASA, Glenn Research Center, Cleveland, Ohio 44135, June 2002.

[86] Willcox, K., "Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition," *Computers & Fluids*, vol. 35, pp. 208–226, February 2006.

[87] Woodbury, K. A., "Method of weighted residuals." Handouts for ME 611 FEA of Convective Heat Transfer.

[88] Yoder, T., "Development of aircraft fuel burn modeling techniques with applications to global emissions modeling and assessment of the benefits of reduced vertical separation minimums," Master's thesis, Massachusetts Institute of Technology, May 2007.

VITA

Kyunghoon Lee was born in Pusan, South Korea, on March 6, 1975. He enrolled in aerospace engineering at Pusan National University (PNU) in March 1994. While he was an undergraduate student, he joined the Republic of Korea Army in January 1996 for obligatory military service, serving as a UH-60P Black Hawk crew chief until he was discharged as a sergeant in March 1998. In February 2001, he received his bachelor's degree in aerospace engineering and subsequently entered the Korea Advanced Institute of Science and Technology (KAIST) for graduate study in the Division of Aerospace Engineering in the School of Mechanical Aerospace and Systems Engineering, but before graduating in 2002, to study abroad. In August 2003, he joined Aerospace Systems Design Laboratory (ASDL) in the School of Aerospace Engineering at the Georgia Institute of Technology and earned his master's degree in December 2004. In the fall of 2005, he passed his Ph.D. qualifying exams in the area of advanced design methods and process, fixed wing design and performance, and aerospace structural analysis, and expects to be awarded his Ph.D. in the summer of 2010. His research interest focuses on utilizing computational data analysis techniques to facilitate aerospace system design and analysis.